Office for
Nuclear Regulation

Validation and Verification of Field Programmable Gate Array based systems

Dr Andrew White

Principal Nuclear Safety Inspector,

Office for Nuclear Regulation, UK

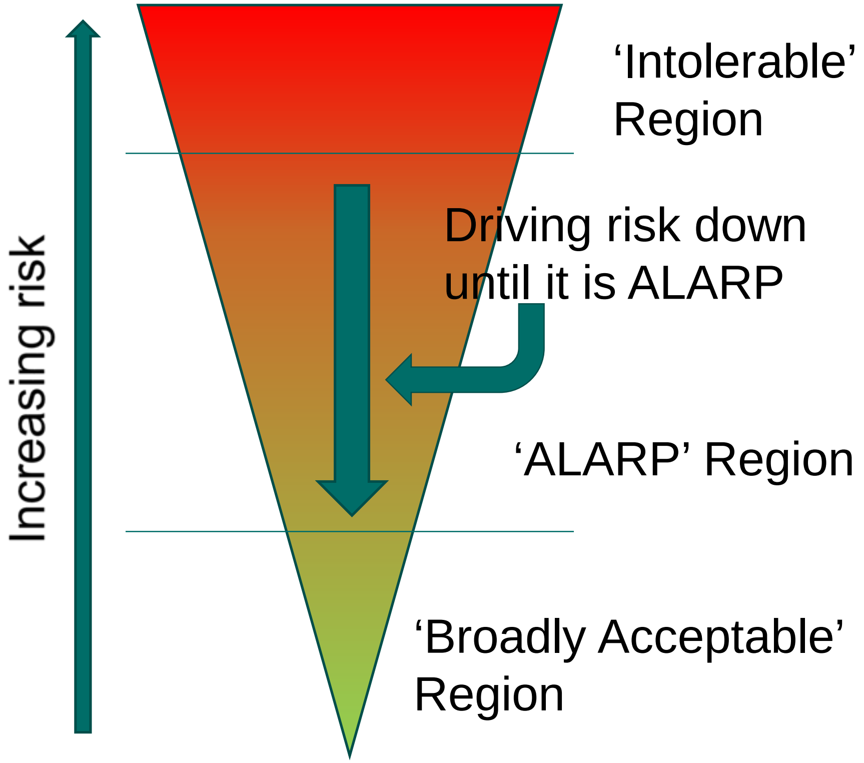
Objectives

- Purpose and activities of the Office for Nuclear Regulation (ONR), and the UK approach to nuclear safety regulation - context
- What are the challenges to ensuring FPGA based systems are adequately reliable?
- The UK approach to managing these challenges
- The bigger picture

The Office for Nuclear Regulation

- ONR regulates the nuclear industry on behalf of the public to ensure that the risks arising from activities in the nuclear industry remain acceptable.
- There is a legal requirement to reduce risk ‘So Far As Is Reasonably Practicable (SFAIRP)’.
- In the UK nuclear industry, we use the term ‘ALARP’ to describe reducing risks to ‘As Low As Reasonably Practicable’.
- SFAIRP and ALARP are used interchangeably

As Low As Reasonably Practicable


Increasing risk
'Intolerable' Region
Driving risk down until it is ALARP
'ALARP' Region
'Broadly Acceptable' Region

- If the 'cost' of a risk reduction measure is grossly disproportionate to the reduction in risk, the risk is considered 'ALARP'
- Practically this is not done through an explicit comparison of cost and benefits, but by applying established relevant good practice (RGP) and standards, and arguing this.

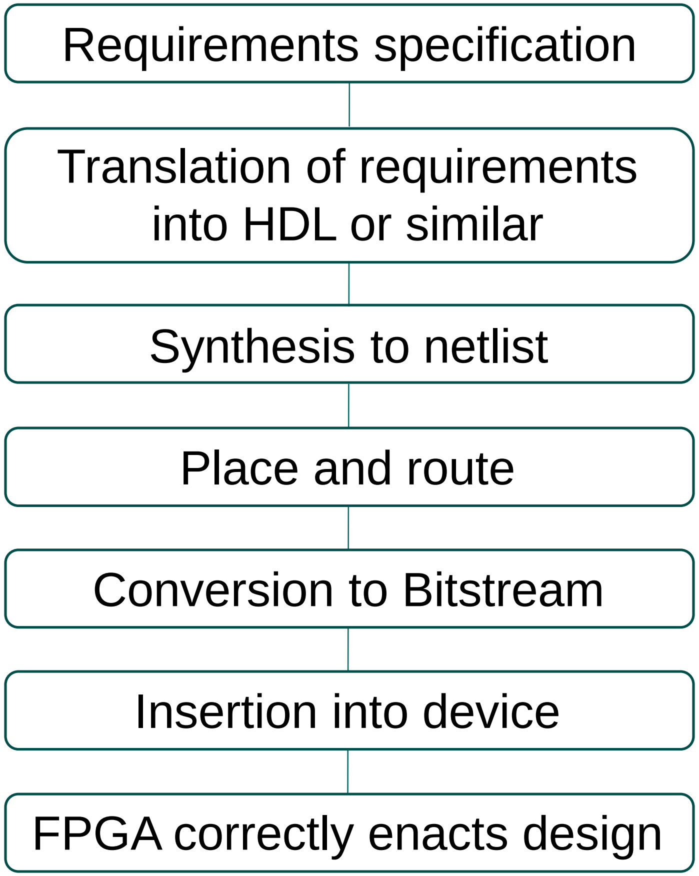
Nuclear regulation in the UK is goal setting

- Licensee’s have to demonstrate they have applied relevant good practice and that risks cannot be further reduced
- There are 36 license conditions that the licensees must adhere to
- Breach of a license condition will result in regulatory action
- The license conditions require that a safety case must be maintained and be a continuous demonstration that activities are being managed so they remain adequately safe

The Safety Case

- Every activity involving nuclear material should have a safety and security case
- This should argue why the risks associated with the activity are ALARP and is often of a Claims, Arguments, Evidence structure
- For this to be successfully argued the potential options for how the activity can be carried out should be described, so that the most appropriate can be selected, and it must be demonstrated that nothing further can be done to reduce risk
- Any modifications to systems or the environment will require the safety case to be updated
- ONR assesses safety cases and requires improvements to engineered systems where the licensee cannot demonstrate that risks are ALARP

Where can faults be introduced into a FPGA design?



Requirements specification

Translation of requirements into HDL or similar

Synthesis to netlist

Place and route

Conversion to Bitstream

Insertion into device

FPGA correctly enacts design

Why is operational experience not sufficient on its own to demonstrate adequate risk control?

- For operational experience to be relevant the device/component has to have been successfully used in a manner that supports the proposed use, including:
- Similar (identical?) use profile
- Configuration (e.g. software/firmware and hardware versions should be the same)
- Any failures have been identified and analysed
- Needs to be statistically significant (e.g. sufficient running hours, demands, etc.)

Why is testing on its own not sufficient to demonstrate adequate risk control?

- For testing to be sufficient all potential internal states need to have been covered
- Even on small systems there are too many internal states (combinations of potential internal memory states) to achieve even 1% of coverage in a reasonable time
- Testing is necessary to demonstrate functional requirements have been met
- Statistical testing provides additional confidence that the system will perform a specific application – see later

What are the challenges with validation and verification of a FPGA design?

Because they are reliant on:

- People
- Software and other engineering tools
- Pre-developed designs e.g. libraries/macros
- The design processes and quality control of other manufacturers
- The design being correctly inserted into the FPGA
- The FPGA correctly enacting the design

ONR's regulation of complex electronic systems

- ONR considers that the failure causes of FPGA's are similar to those of microprocessor-based systems, namely:
    - Incorrect/inadequate requirements specifications – at system and module level
    - Unsuitable/inadequate design and development processes
    - Design decisions that result in inadequate architectures – at system and module level
    - Inability to fully analyse/test the design due to its complexity
    - Inadequate/ineffective validation and verification processes

Failure to maintain focus on the desired safety properties

ONR's expectations for demonstrations of adequacy for complex systems

- ONR Technical Assessment Guide (TAG) 46 "Computer based safety systems" describes how risks arising from computer based systems should be managed.
- This expects the safety case argument to consist of two 'legs':
  - Production excellence
  - Independent confidence building
- Both legs need to be sufficiently strong to make an adequate case for safety (i.e. one leg only is not sufficient)

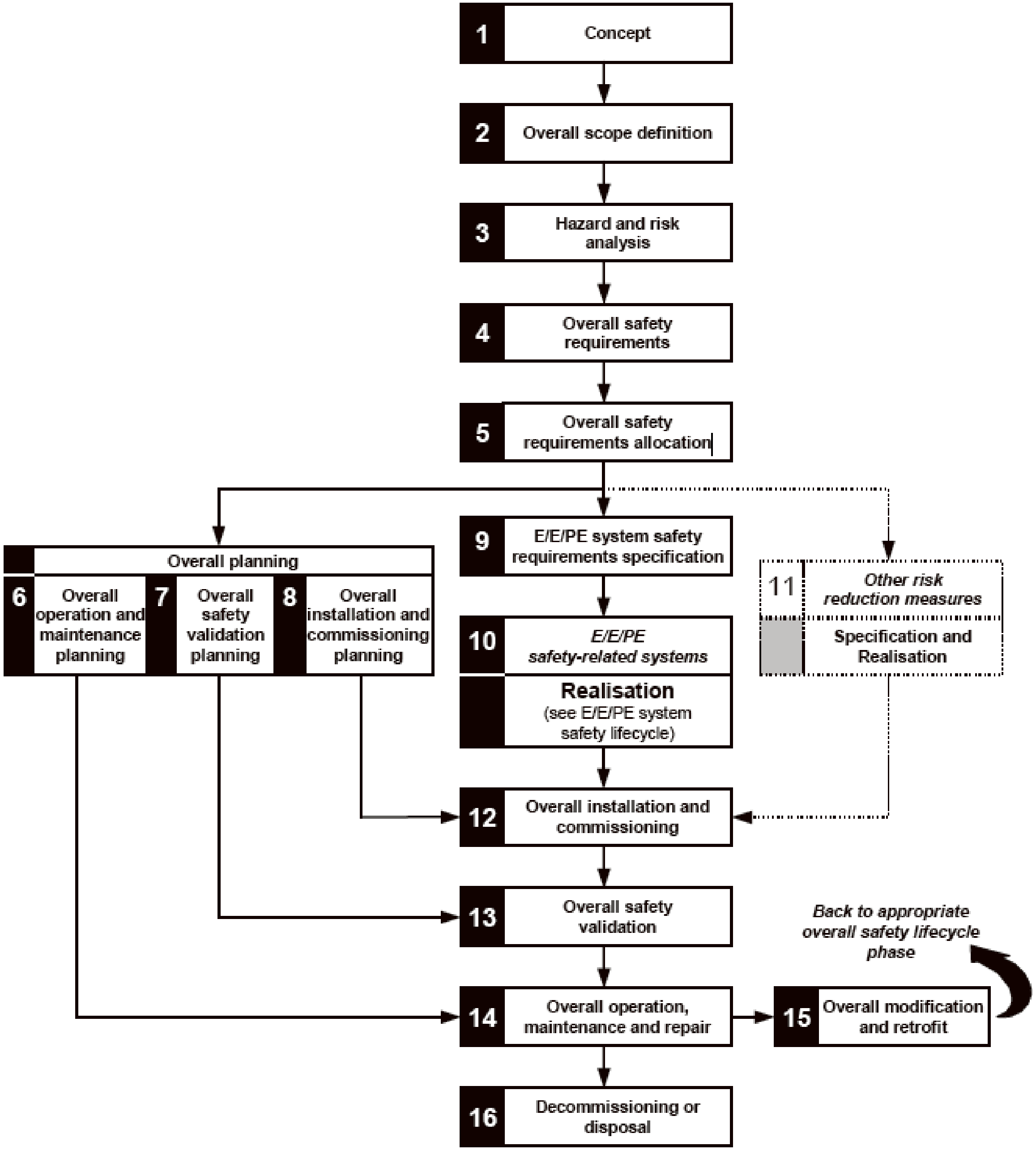
Production Excellence

- In the UK we refer to international standards, e.g. IAEA and IEC standards for production excellence.
- Standards include:
- IAEA Safety Standards Series, Specific Safety Guide No.SSG-39 – Design of Instrumentation and Control Systems for Nuclear Power Plants.
- IAEA NP-T-3.17 “Application of Field Programmable Gate Arrays in Instrumentation and Control Systems of Nuclear Power Plants

Production Excellence – examples of standards

- IEC 61508 Functional safety of electrical electronic programmable electronic safety-related systems
- IEC 61513 Nuclear power plants — Instrumentation and control for systems important to safety
- IEC 62566 Nuclear power plants — Instrumentation and control important to safety — Development of HDL-programmed integrated circuits for systems performing category A functions
- IEC 60880 Nuclear power plants - Instrumentation and control systems important to safety - Software aspects for computer-based systems performing category A functions

The 61508 safety lifecycle


1 Concept
2 Overall scope definition
3 Hazard and risk analysis
4 Overall safety requirements
5 Overall safety requirements allocation
Overall planning
6 Overall operation and maintenance planning
7 Overall safety validation planning
8 Overall installation and commissioning planning
9 E/E/PE system safety requirements specification
10 E/E/PE safety-related systems
Realisation
(see E/E/PE system safety lifecycle)
11 Other risk reduction measures
Specification and Realisation
12 Overall installation and commissioning
13 Overall safety validation
14 Overall operation, maintenance and repair
15 Overall modification and retrofit
Back to appropriate overall safety lifecycle phase
16 Decommissioning or disposal

Independent Confidence Building Measures

- Includes activities that may not be a part of production excellence such as:
- Code review
- Concurrency analysis
- Dynamic code analysis
- Static code analysis
- Statistical testing

- In order to provide added confidence the production process has produced a module/system of sufficient reliability

However, there are other things that are important to complete the safety demonstration

Relating to:

- The use of unverified/malicious code
- The suitability of the FPGA for its environment
- Appropriate use of complex (unverifiable) functionality within the FPGA
- The potential for software tools to contain faults

Use of libaries, macro’s, predeveloped designs

- Does the predeveloped design come from a trusted source?
- What verification has been performed on it?
- Could it contain malicious code?
- Can you verify it?
- If the pre-developed was to contain malware, what effect could it have?

Types of FPGA design technology

Commonly used are:

- Fuse/Anti fuse
- Static ram
- Flash

Vulnerabilities of flash/SRAM technology

- Single event upsets (SEU's) should be considered in the design, particularly if there is a requirement to operate in significant radiation fluxes
- Countermeasures include:
  - internal design (within the FPGA) to detect this and prevent an erroneous output
  - Module design, comparing outputs of devices performing same function
  - At system level – e.g. a four division voted architecture where no single failure leads to a loss of the safety function

Use of complex functionality in the FPGA including

- Microprocessor cores
- Communication processors
- Memory management, and other complex functions

- To what extent have these functions been verified and how?

Software Tools for FPGA’s

- There is the potential for software tools to contain faults that could result in a safety consequence. This may be addressed by a number of different approaches:
  - Use of proven in use tools. This is vulnerable to version changes
  - Certification of tools. This is vulnerable to version changes
  - Use of diverse tools and cross compare. Noting some tools may have a common history
  - Assessment of the effects of a fault in a tool, and taking action to add an independent check, or mitigation
  - Use of formal methods to formally prove the correctness of the design at each stage

Bitstream

- Bitstream is generally encrypted – how is it possible to know the bitstream reflects the correct design?
- Has the design been correctly transmitted to the device?
- Are all gates correctly programmed?
- Is there any unwanted functionality?
- Can the design be read back from the device?

A word on Statistical Testing

- Statistical testing is a mathematically based testing technique that can give an estimate of probability of failure on demand for a demand based system.
- The system is tested with a large number of demands that reflect the demand profile for the system
- The system is reset to a known state between tests so that the tests are statistically independent
- 50,000 tests with no failures provides 99% confidence that a $1x10^{-4}$ probability of failure on demand has been achieved.
- However, there are some health warnings – see next slide:

A word on Statistical Testing

- The test coverage of total system states remains miniscule
- If the actual input profile during use is different to the demand profile used for the tests, then the reliability claim cannot be maintained
- If there is any test failure, the system needs to be fixed, and the cause of the failure needs to identified, including why this was not identified by the verification and validation measures
- Statistical testing cannot identify ‘creeping death’ failures such as a gradual inability to process inputs caused by undetected failures/loss of system resources

Wider systems issues

- Is there diversity between layers of protection? For example is the reactor control system microprocessor based, and the protection system FPGA based?
- Are there still common components (e.g. analogue to digital convertors) shared across layers that could fail in the same way at the same time?
- Are sensor inputs shared between layers of protection?
- Are the different layers of protection dependent upon the same support systems e.g. electrical power, cooling (HVAC), instrument air, etc.

Wider systems issues

- Can lower class systems prevent actions of higher class systems through priority actuation systems?
- Is there communication from lower classified systems to higher classified systems?
- How is the potential for spurious actuation being considered?
- Is there a common maintenance regime?
- Is the resistance to common cause faults similar in other technology systems (e.g. mechanical systems)?

Conclusion

- Incorrect or misinterpreted requirements cannot be corrected by any verification or validation technique
- Any high complexity system that is intended to achieve a high reliability requires a full range of techniques to ensure that faults arising from every stage of the lifecycle are eliminated, mitigated and reduced
- FPGA based systems have the potential to provide high reliabilities due to their inherent design constraints
- No single technique can eliminate all (or even the majority of) faults, although some techniques are very powerful at doing this, such as the use of formal methods

Conclusion (continued)

- If the architecture is wrong, it is very difficult to produce a high reliability system and to demonstrate it is high reliability
- It is essential that in any verification and validation activity that the focus is on what faults it can detect, and what it can't. Be clear what can be detected and what cannot
- Any system is only as reliable as the electromechanical system to which it is connected. Things such as poor maintenance and incorrectly positioned sensors will significantly affect system reliability

Office for
Nuclear Regulation

Questions?