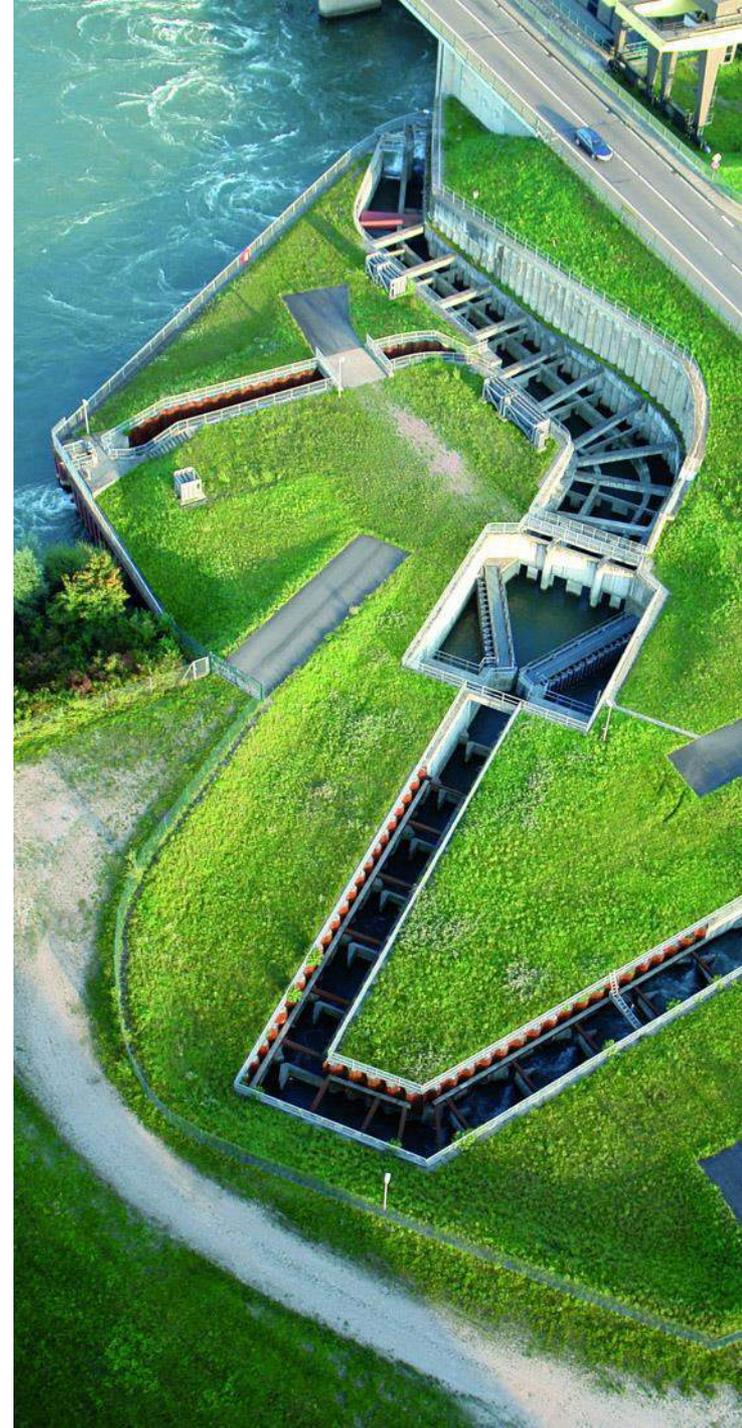# NPIC HMIT 2015

## FPGA-Based I&C Systems : Unraveling Myths from Reality

EDF Nuclear Engineering Division, Basic Design (SEPTEN)

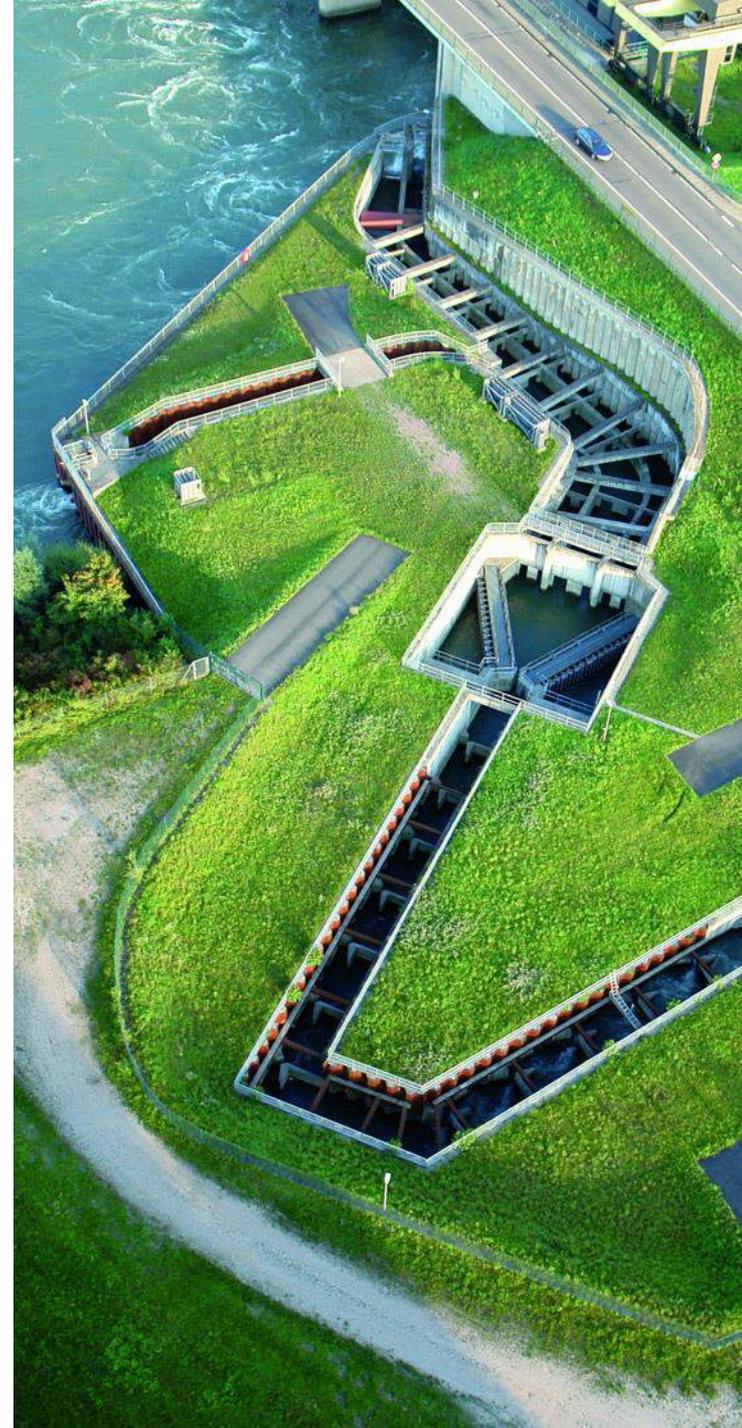Alexander Wigg (alexander-john.wigg@edf.fr)
Ludovic Pietre-Cambacedes (ludovic.pietre-cambacedes@edf.fr)

# Presentation Plan :

1. Introduction

2. Key differences between FPGAs and microprocessors

3. Thematic analysis

4. Conclusions and recommendations

# Introduction and objective

- Objective : To promote critical analysis and open, non-biased discussion based on facts and real world experience.

# Introduction and objective

The problem with how FPGAs are regarded in the nuclear industry :

| Theme | In the beginning… | Now… |
|---|---|---|
| Qualification | No software so easy to qualify | Difficult to qualify, but at least there is no operating system with FPGAs |
| Cybersecurity | Immune to viruses | Viruses are not the only issue, but FPGAs have certain advantages |
| Deterministic behaviour | Pure hardware solution so deterministic behaviour | Pure hardware solution so deterministic behaviour |
| Performance | Faster response times | Faster response times |
| Obsolescence | Transferrable code, no obsolescence issues | Code is transferrable, but FPGAs do, in fact, become obsolete |

# Introduction and objective

The problem with how FPGAs are regarded in the nuclear industry :

| Theme | In the beginning… | Now… |
|---|---|---|
| Qualification | No software so easy to qualify | Difficult to qualify, but at least there is no operating system with FPGAs |
| Cybersecurity | Immune to viruses | Viruses are not the only issue, but FPGAs have certain advantages |
| Deterministic behaviour | Pure hardware solution so deterministic behaviour | Pure hardware solution so deterministic behaviour |
| Performance | Faster response times | Faster response times |
| Obsolescence | Transferrable code, no obsolescence issues | Code is transferrable, but FPGAs do, in fact, become obsolete |

**<u>Presented in this manner, the offered "advantages" in fact have very little to do with FPGA technology.</u>**

# Key differences between FPGAs and microprocessors

- Whether the fundamental distinction between hardware and software matters or not will need to be determined through discussions between equipment suppliers, plant operators and regulatory bodies.

- The fundamental differences in operating principles can however have an impact on the choice of technology for a particular application.

# Thematic Analysis – Simplicity and deterministic behaviour

- It is true that FPGA-based platforms do not use operating systems...

- …But neither do their "rival" software platforms (i.e. 1E platforms).

- The absence of 'operating systems' or 'unused functionalities' is a requirement according to IEC standards, regardless of the technology.

- Software-based 1E nuclear I&C platforms are completely different to COTS platforms.

**Software ≠ Operating System**

- Arguments for simplicity seem to be based on an older and idealistic understanding of FPGAs.

- In complex FPGAs, poorly designed clock signals can result in unpredictable behaviour, even in synchronous designs.

**Pure hardware ≠ deterministic behaviour**

# Thematic Analysis – Obsolescence

- The COTS market drives functionality and the need for complexity .

- COTS suppliers also *want* their products to become obsolete.

- This makes COTS generally unsuitable for nuclear applications.

- This mechanism applies to microprocessors **and** to FPGAs.

**The same principles apply for nuclear products, although cycles are longer**

- In modern FPGAs, the use of vendor-specific functionalities is inevitable for all but the simplest of functions.

- The utility will typically have no influence in how code is written, how portable it is, nor will they have the rights to implant it on different hardware, even if it is possible.

**Software and HDL can <u>both</u> be designed to be portable**

# Thematic Analysis – Cybersecurity

| Common Argument | Analysis |
|---|---|
| *"FPGA-based systems are less vulnerable to cyber-attacks than microprocessor-based ones"* | For attacks targeting executable code, FPGAs may have some advantages, but other attacks are in fact much more common. |
| *"FPGA-based systems can be designed without high-level, general purpose components which are easily attacked"* | No 1E platform is designed using general purpose components. |
| *"FPGA re-programming can be possible only by physical access, or anti-fuse FPGAs can be used"* | A software platform could also be designed in this way and unfortunately no FPGA platforms use anti-fuse technology. |
| *"IP cores can be verified to be free of hidden or unnecessary capabilities"* | What is true for HDL is also true for software. For class 1, black box components are prohibited. For class 2 or 3, which may use COTS, access may not be allowed, whether it's software or HDL. |

# Thematic Analysis – Licensing and qualification

**Problem : How to treat FPGAs compared to software?**

- In France, software-based protection systems were qualified well before appropriate software standards existed.

- Such real-world experience is essential for the development of relevant and industrially applicable standards.

- The need for software standards arose from increasing complexity of systems.

**Therefore, the need for FPGA standards is a result of their complexity, not of their "resemblance" to software, despite the fact that they are in the end purely hardware.**

# Conclusion and recommendations

- In COTS components, FPGAs and microprocessors are used concurrently in a complementary fashion. FPGAs are fundamentally better for some applications (parallel high-speed processing), and microprocessors for others (floating point).

- Software-based safety platforms also use FPGAs for some peripheral tasks, and vice-versa, due to their suitability for them.

**FPGAs and software are not rivals, they are different.**

- Improved portability, simplicity and deterministic behaviour: these arguments are based on an older and idealistic understanding of FPGAs, and perhaps a lack of knowledge of current software-based safety platforms, which bear little resemblance to the COTS with which FPGAs are so often compared.

- The choice of technology alone tells you very little about your final system.

**The decision to use FPGAs or not should be based more upon the functional requirements of the system, and less upon the hope of them being an easier option to qualify than software.**

# Conclusion and recommendations

| | In the beginning… | Now… | After analysis |
|---|---|---|---|
| Qualification | No software so easy to qualify | Difficult to qualify, but at least there is no operating system with FPGAs | Technology independent. Design dependent |
| Cybersecurity | Immune to viruses | Viruses are not the only issue, but FPGAs have certain advantages | Technology independent to an extent. Design dependent |
| Deterministic Behaviour | Pure hardware solution so deterministic behaviour | Pure hardware solution so deterministic behaviour | Technology independent. Design dependent |
| Performance | Faster response times | Faster response times | Technology independent. Design dependent. |
| Obsolescence | Transferrable code, no obsolescence issues | Code is transferrable, but FPGAs do, in fact, become obsolete | Technology independent. Design dependent. |

# Thank you for your attention.

EDF Nuclear Engineering Division, Basic Design (SEPTEN)

Alexander Wigg (alexander-john.wigg@edf.fr)
Ludovic Pietre-Cambacedes (ludovic.pietre-cambacedes@edf.fr)