

# Bio - Kook Hun Kim

## ■ Senior Member, National Academy of Engineering of Korea



- Name : Kook Hun KIM
- Phone : +82-10-5492-1440
- E-Mail : kerikim0328@daum.net

### Education and Careers

- Academician of Korea since 2012 (National Academy of Engineering of Korea)
- DG of KNICS R&D Center (~2008)
- SVP and Head of Nuclear I&C BU, DHIC (2008~2017)
- Senior/Principal Researcher in Korea Gov. Institute, (KERI, 1989~2008)
- Post Doctoral Research (U. of Oxford, 1988)
- BS, MS and PhD from SNU, EE (1979, 1983, 1987)

### R&D Achievements

- Korean Nuclear I&C System(KNICS)
- Nuclear I&C Business (QA/QC Set-Up, Manufacturing, Supplying, SC, EQ) and project execution
- Nuclear I&C Global Cooperation
- IAEA's Expert Activities
- Dual Redundant and Triple Redundancy Exciter
- Design of Signaling System for High Speed Train
- Korean 1st Generation Distribution Automation System(Central Control System, Basis of Smart Grid)
- MAGLEV Control System

### Awards

- Developer Award of One of the 100 Key Technologies since 1948, Korea (Dec. 2011)
- Korea Industrial Medal (Dec. 2012)
- Korea Scientific & Technology Medal (Apr. 2001)
- Scientist and Engineer Award of the Month, Korea (Oct. 1999)
- Many Awards from Academic Institutes (KIEE, ICROS)

# **Development and Application Experience of FPGA and CPU based system for safety related I&C system**

**11<sup>th</sup> International FPGA Workshop  
October 8-11, 2018 in Dallas, Texas**

**Kook Hun Kim,  
Ph. D  
Academician, National Academy of Engineering of Korea**

## 1. Exploring the experience of Korea

- PLC
- V&V Activity for PLC
- FLC(FPGA Logic Controller)
- V&V Activity for FLC

## 2. Comparison analysis

- Lifecycle
  - ✓ Platform, Application and IEEE 1012 vs IEC 62566
- Feedback from Korean Experience(CPU and FPGA)
  - ✓ Regulator, Designer & Developer and V&V

## 3. Conclusion

# PLC – Modules configuration

## ❑ Processor Module

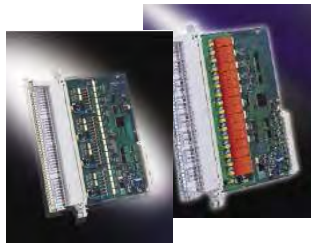
- Single Processor Module
- Redundant Processor Module

## ❑ Communication Module

- HR-SDL Module
  - Trip signals between channels for RPS
- HR-SDN Module
  - Safety control signals for ESF-CCS

## ❑ I/O Module

- Digital Input
- Digital Output
- Special Module

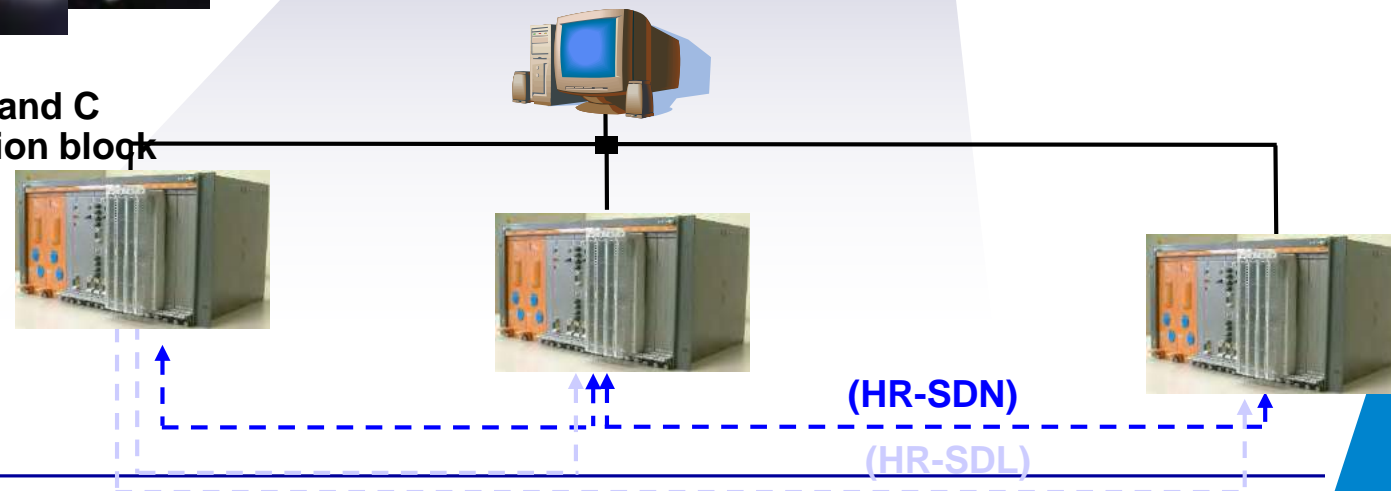


## ❑ Engineering Tool (pSET)

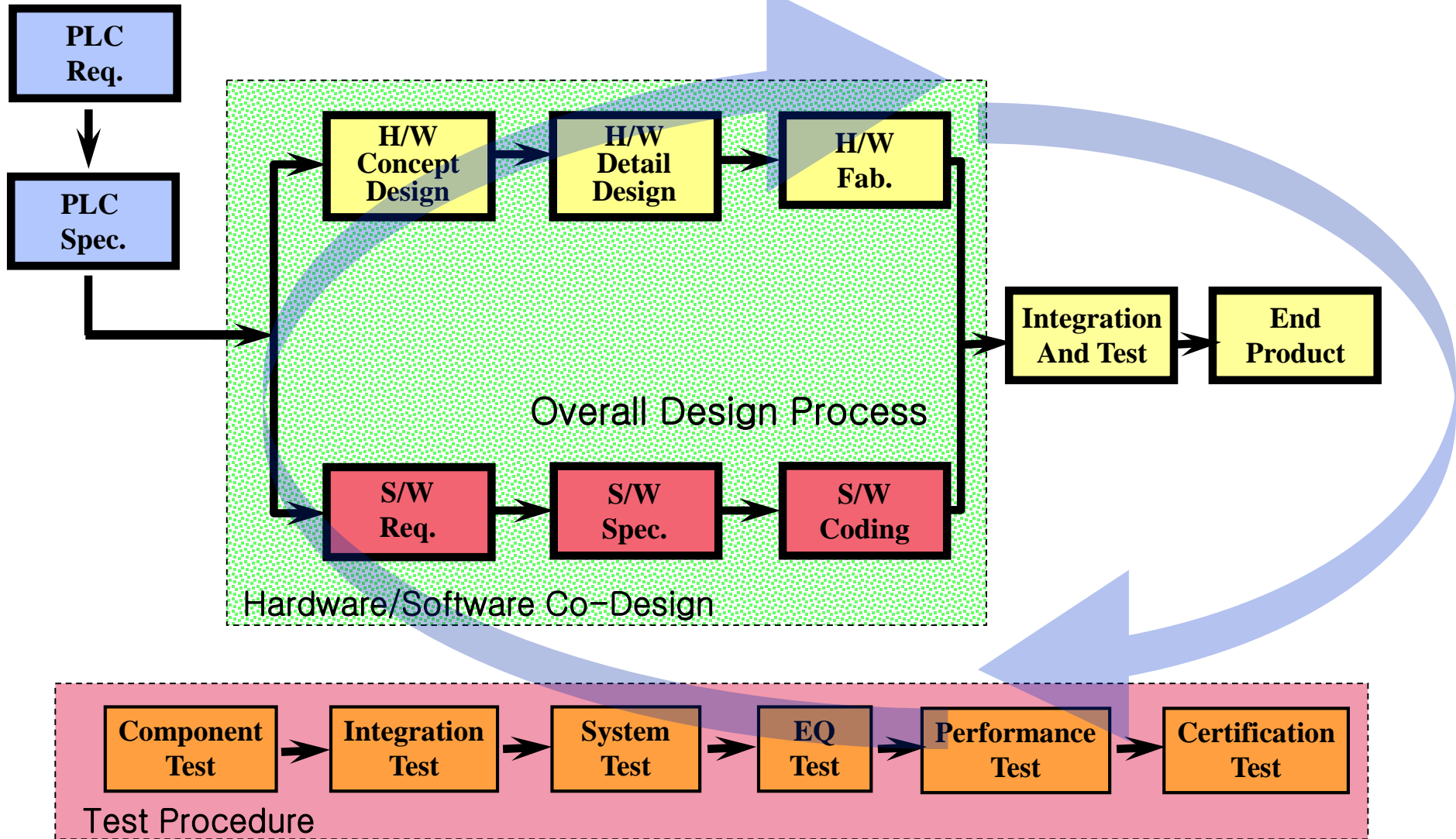
- Support function block and C language for user function block



482.6 x 281.35 x 294mm  
(19 inch Standard)

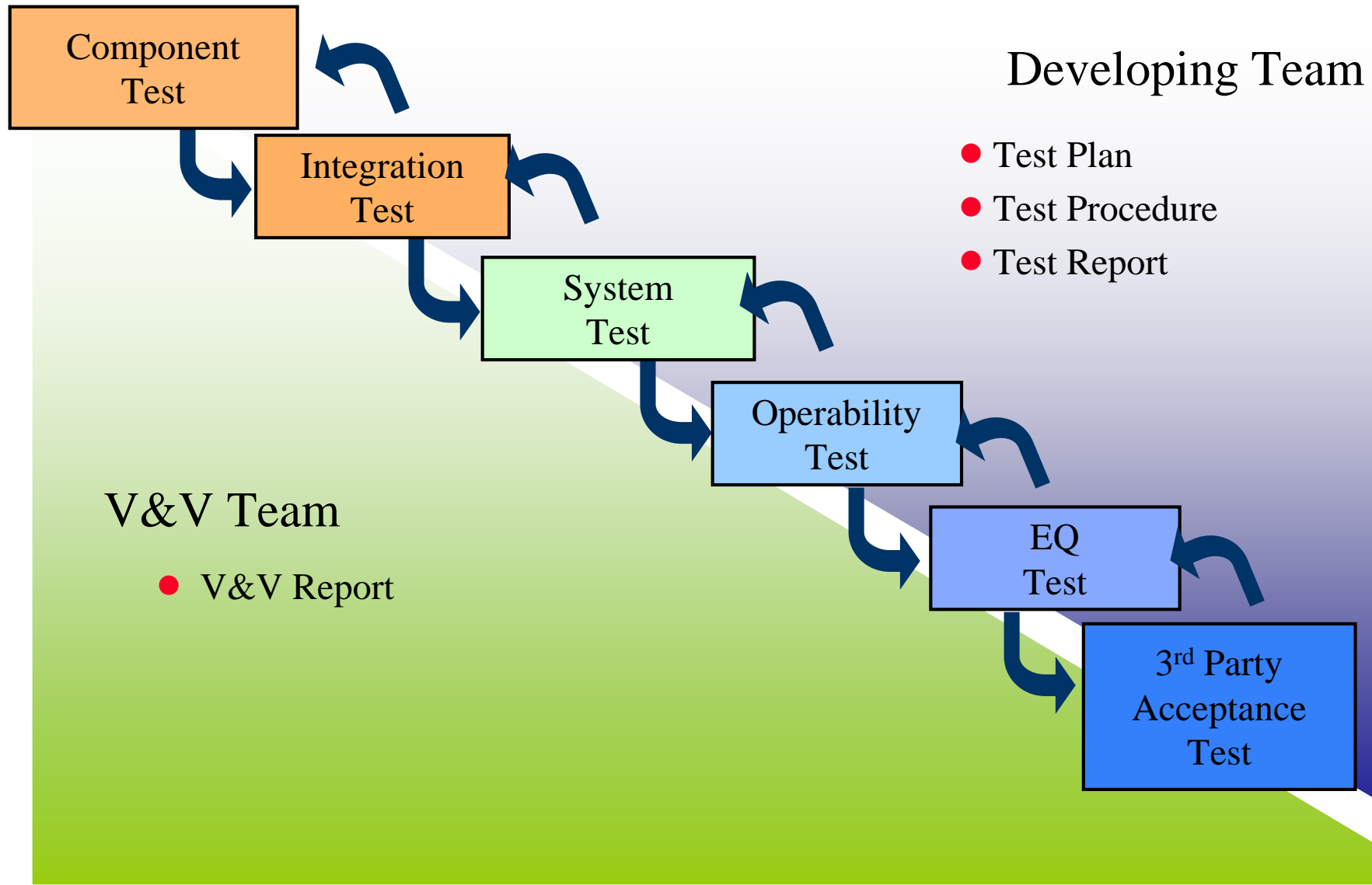


# PLC - Overall Design Process



- ❑ **Design According to S/W Design Plan and Procedure**
  - **USNRC Reg. Guide 1.173**
  - **IEEE Std. 1074**
- ❑ **Development Apply to Structural development method**
  - **Formal method based SRS/SDS (State Chart, SDL)**
- ❑ **Verification According to S/W V&V Plan and procedure**
  - **USNRC Reg. Guide 1.172**
  - **IEEE Std. 1012**
  - **Independent reviewer**
    - **Independent review team in KAERI and V&V specialty company : All S/W**
    - **iSTec : RTOS, HR-SDL S/W**
- ❑ **Test According to Test Plan and Procedure**
  - **Component test**
  - **Integration test**
  - **System test**

# PLC – Test Process



# PLC – Component Test

## □ Purpose

**: H/W & S/W component function/performance Test**

## □ Test Items

### ● Hardware component

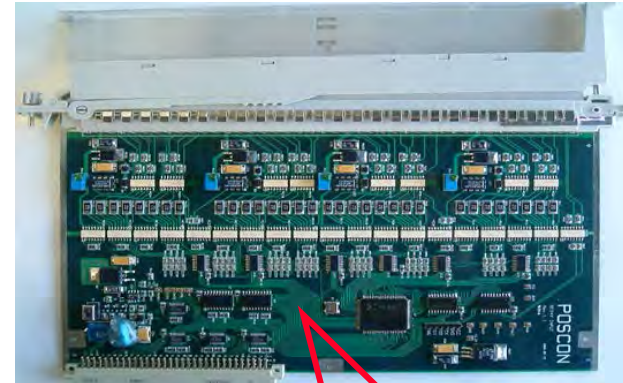
#### ■ Unit H/W modules

- Timer
- Amplifier circuit
- Watchdog timer circuit
- A/D and D/A convert circuit
- Loopback circuits
- LED circuits, etc.

### ● Software component

#### ■ Unit S/W subroutine (sub-functions)

- Black box Test : External check of the subroutine
- White box Test : Internal check of the subroutine



- H/W Component  
- S/W Component



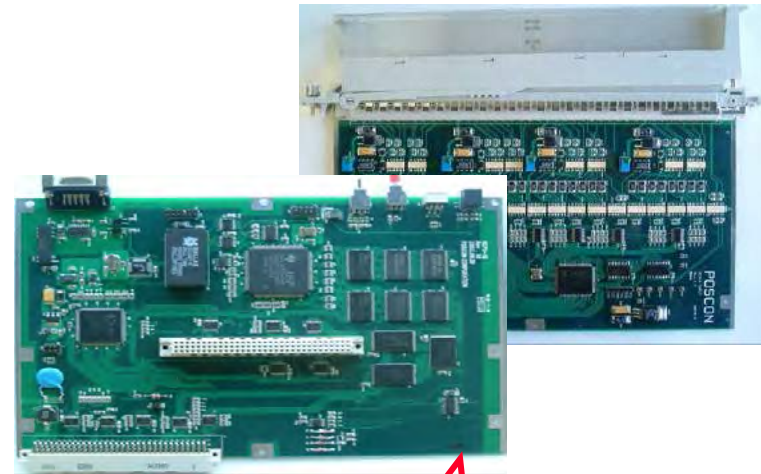
# PLC – Integration Test

## □ Purpose

**: Integrated performance test for H/W & S/W components**

## □ Test Items

- **H/W Integration Test**
  - Integrated test for unit H/W
- **S/W Integration Test**
  - Integrated test for unit S/W
- **H/W and S/W Integration Test**
  - Integrated test for H/W and S/W
- **Integration with Processor Module**
  - Integrated test with processor module



- H/W + S/W Component  
- Integration with Processor  
Module

# PLC – System Test

## ❑ Purpose

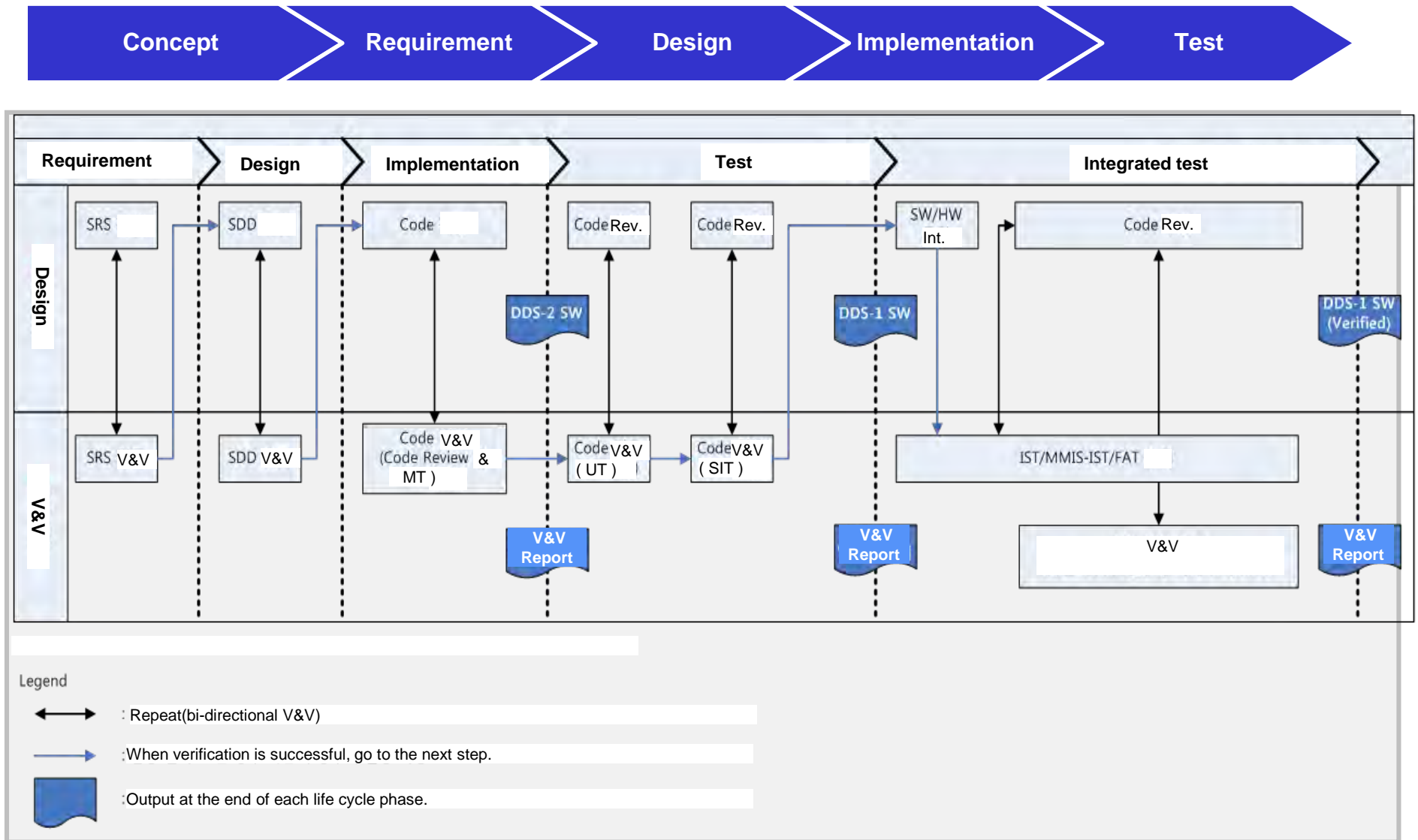
**: Performance test for safety  
system platform**

## ❑ Test Items

- Response time test
- I/O capabilities test
- Memory capacity and data retention capability test
- HR-SDL performance test
- Error Handling capability test
- Max. power supply capability test



# V&V Activities for PLC application – flow chart



# V&V Activities for PLC application – Code Review

Concept

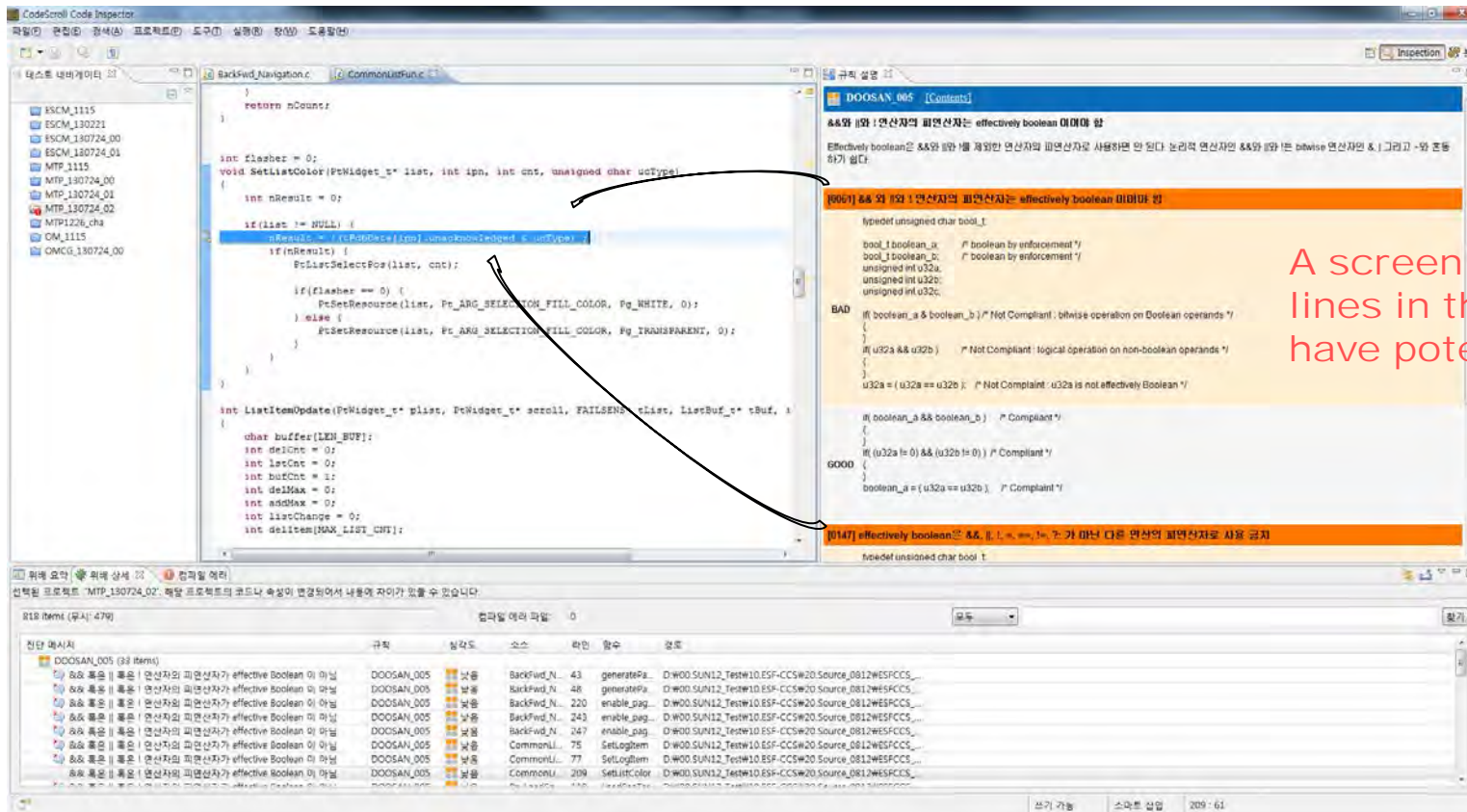
Requirement

Design

Implementation

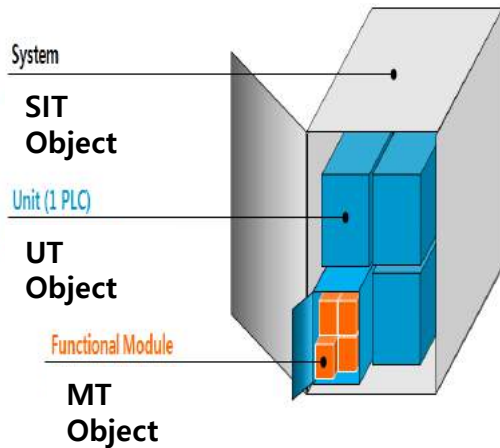
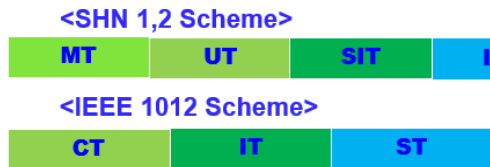
Test

Code Scroll, a source code analysis tool, is applied to prevent human errors and improve efficiency in the review process.



A screen showing which lines in the source code have potential errors

# V&V Activities for PLC application – V&V Test



- ◎ MT : Software Module Test
- ◎ UT : Software Unit Test
- ◎ SIT : Software Integration Test
- ◎ V&V : Verification and Validation
- IST: DHIC (total system test)

Test	Definition and purpose	Scope
MT	<ul style="list-style-type: none"> <li>• Tests to ensure that the Software Functional Module meets the Software Design Specification (SDD).</li> </ul>	<ul style="list-style-type: none"> <li>• S/W Functional Module(Detailed module for each function constituting S / W Unit)</li> </ul>
UT	<ul style="list-style-type: none"> <li>• Test to ensure that the integrated software Functional Module (S / W Unit) meets the Software Requirements (SRD).</li> </ul>	<ul style="list-style-type: none"> <li>• S/W Unit (S/W unit mounted on each controller)</li> </ul>
SIT	<ul style="list-style-type: none"> <li>• Test to confirm that the software installed in the unit system meets the software requirements (SRS).</li> </ul>	<ul style="list-style-type: none"> <li>• System application S/W</li> </ul>

# V&V Activities for PLC application – ESF-CCS IST

Concept

Requirement

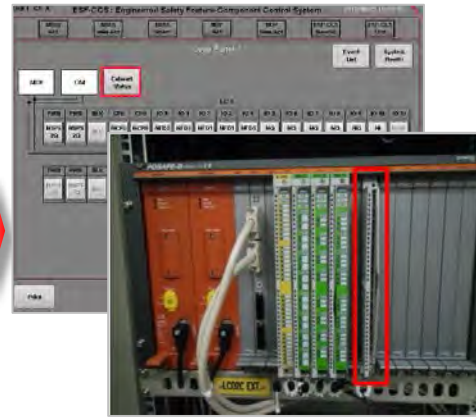
Design

Implementation

Test



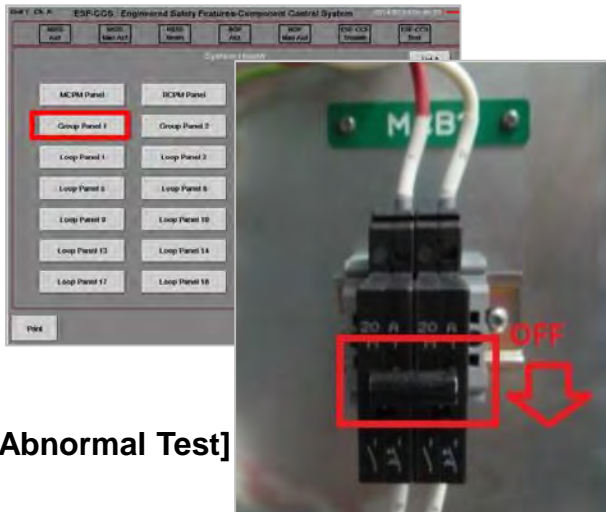
[Power Test]



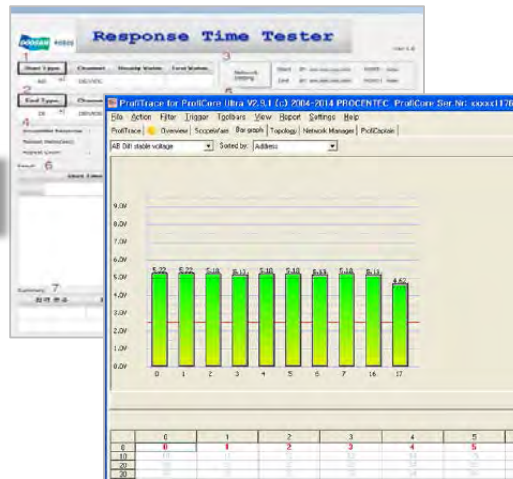
[Cabinet status indication test]



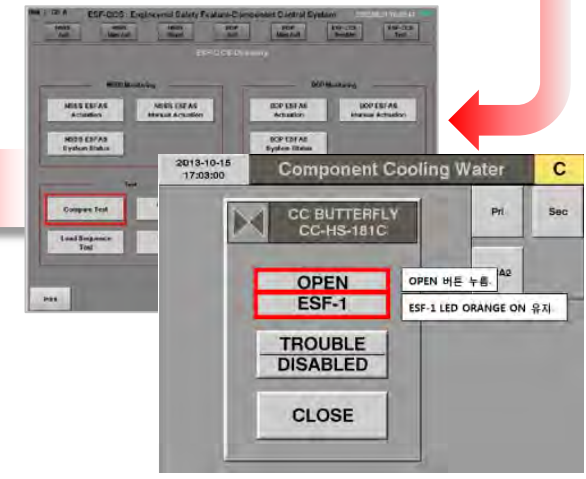
[I/O accuracy test]



[Abnormal Test]



[Performance test]



[Functional test]

# V&V Activities for PLC application – Full scope Integrated Test

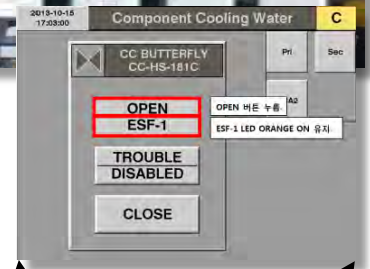
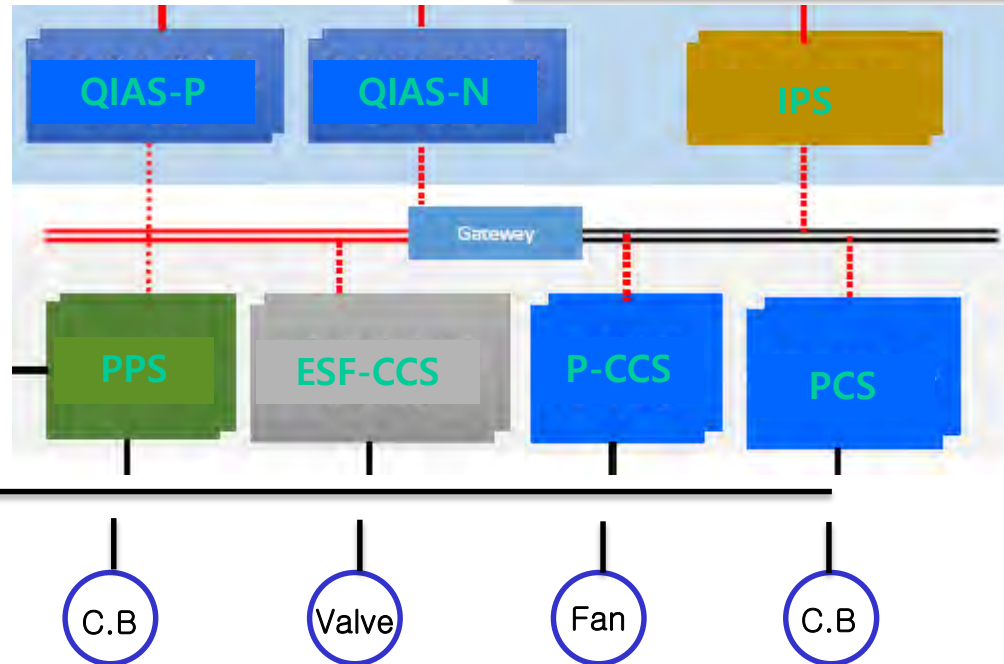
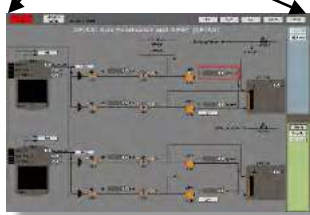
Concept

Requirement

Design

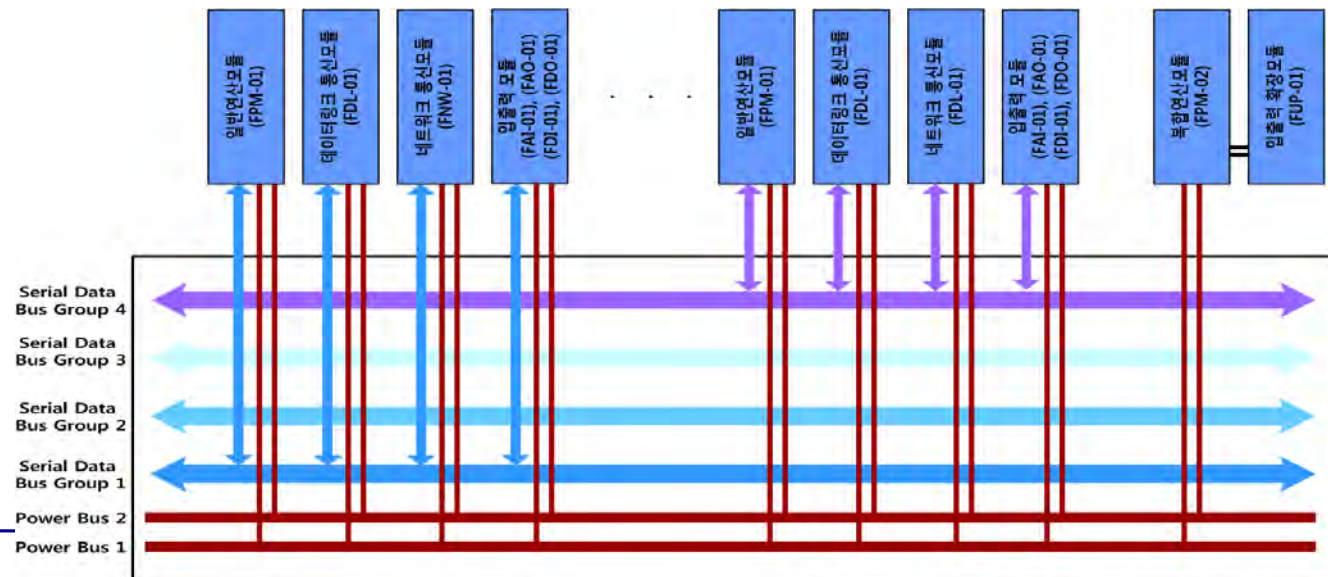
Implementation

Test



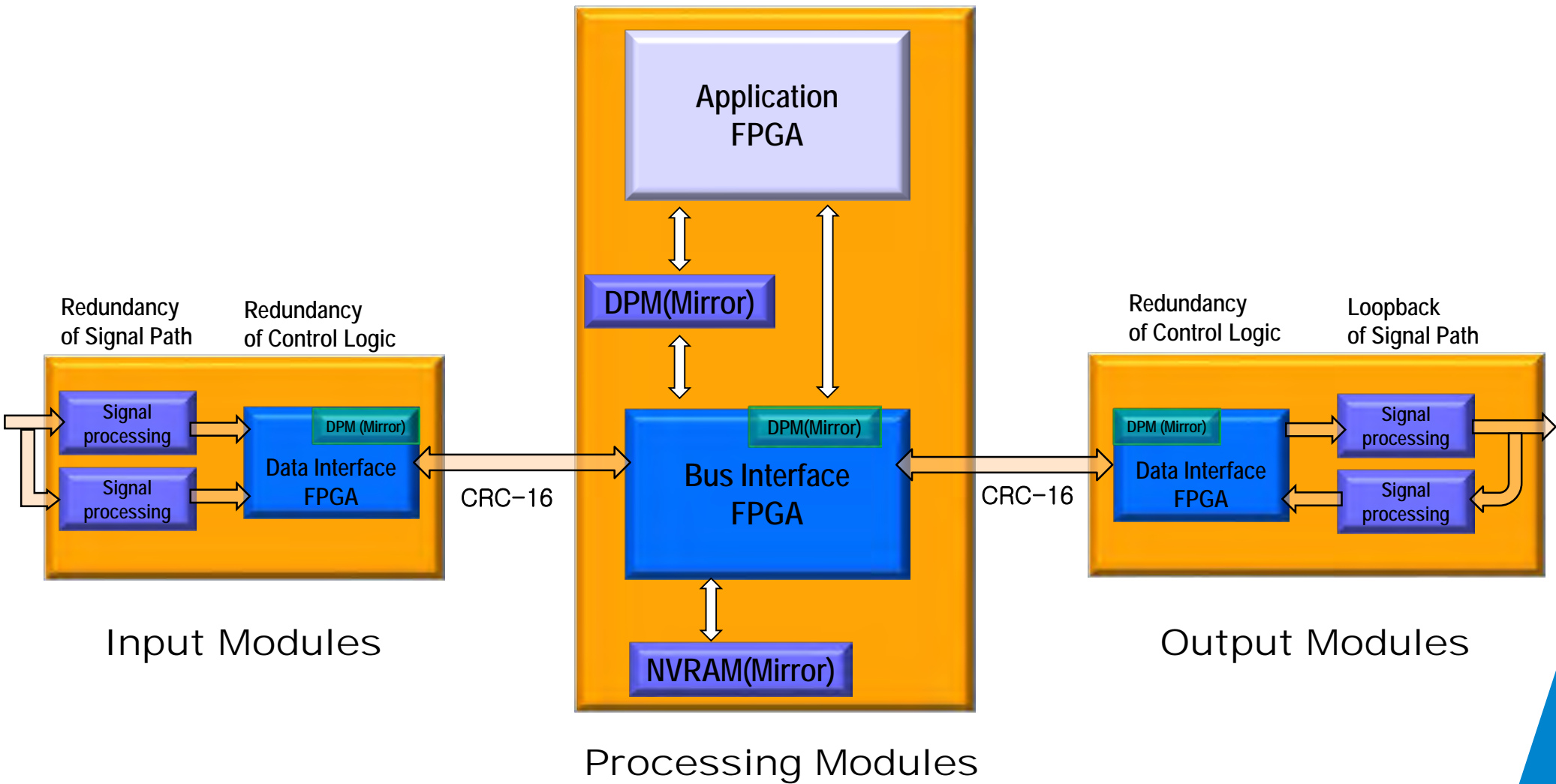
# FLC(FPGA Logic Controller) – Modules configuration

- Data bus architecture for 4 independent multi-processing in one subrack
- 2 redundant power bus
- Processor Module
- Communication Module
  - Safety Data Link
  - Information Network
- I/O Modules
  - Analog I/O Modules
  - Digital I/O Modules
- 482.6 x 281.35 x 294mm (19 inch Standardization)

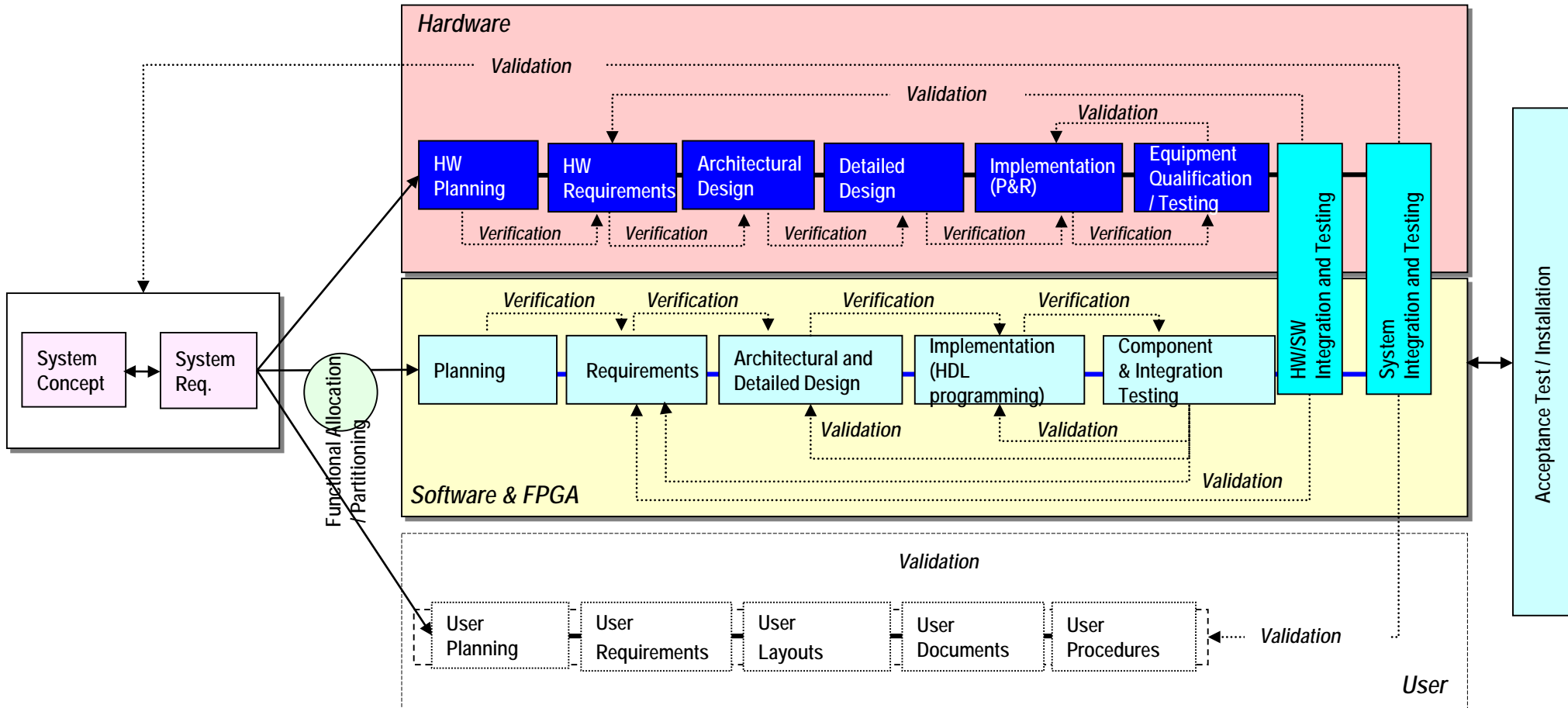




# FLC - Overall system diagnostics

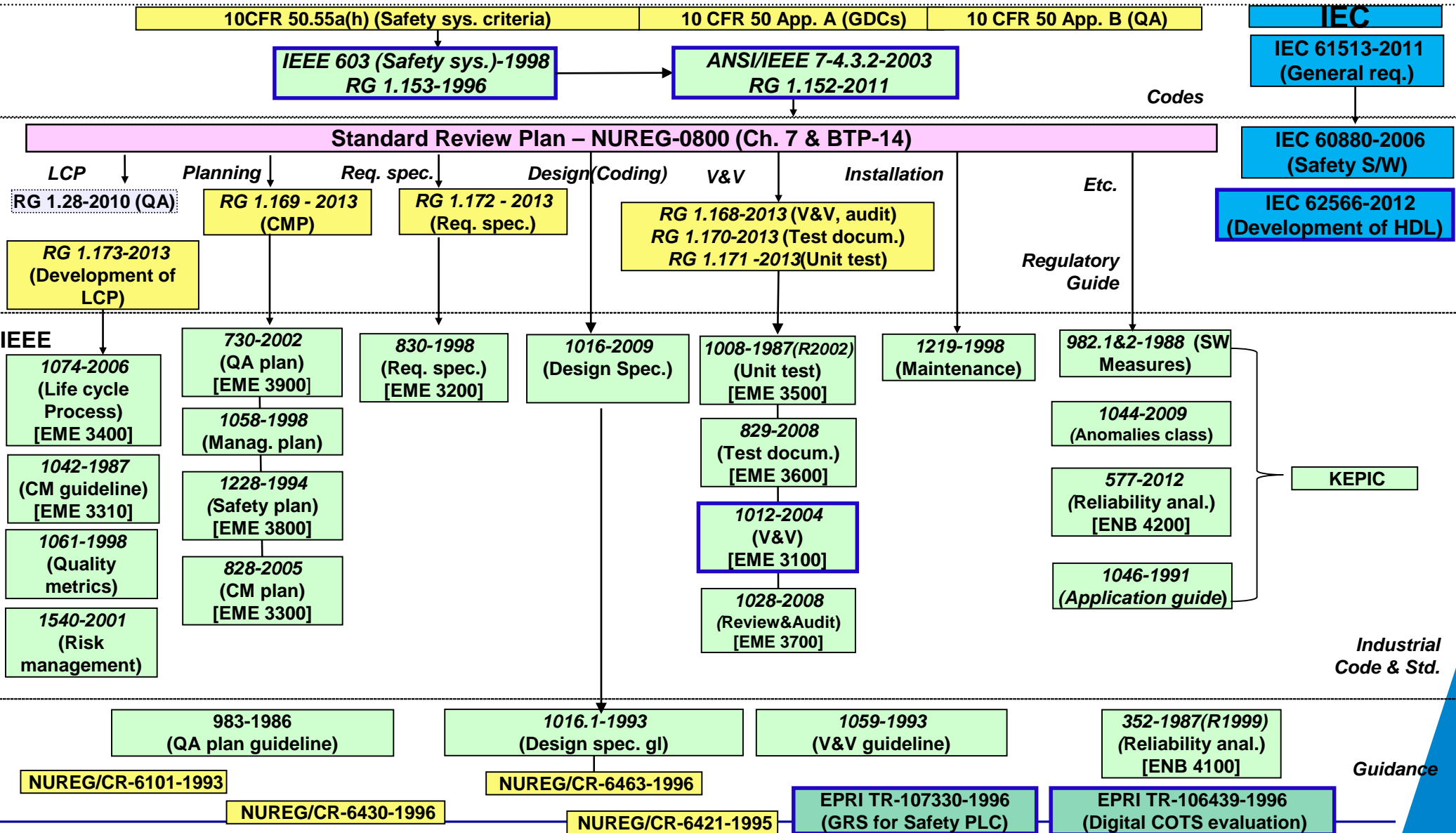


# FLC- Development Life cycle

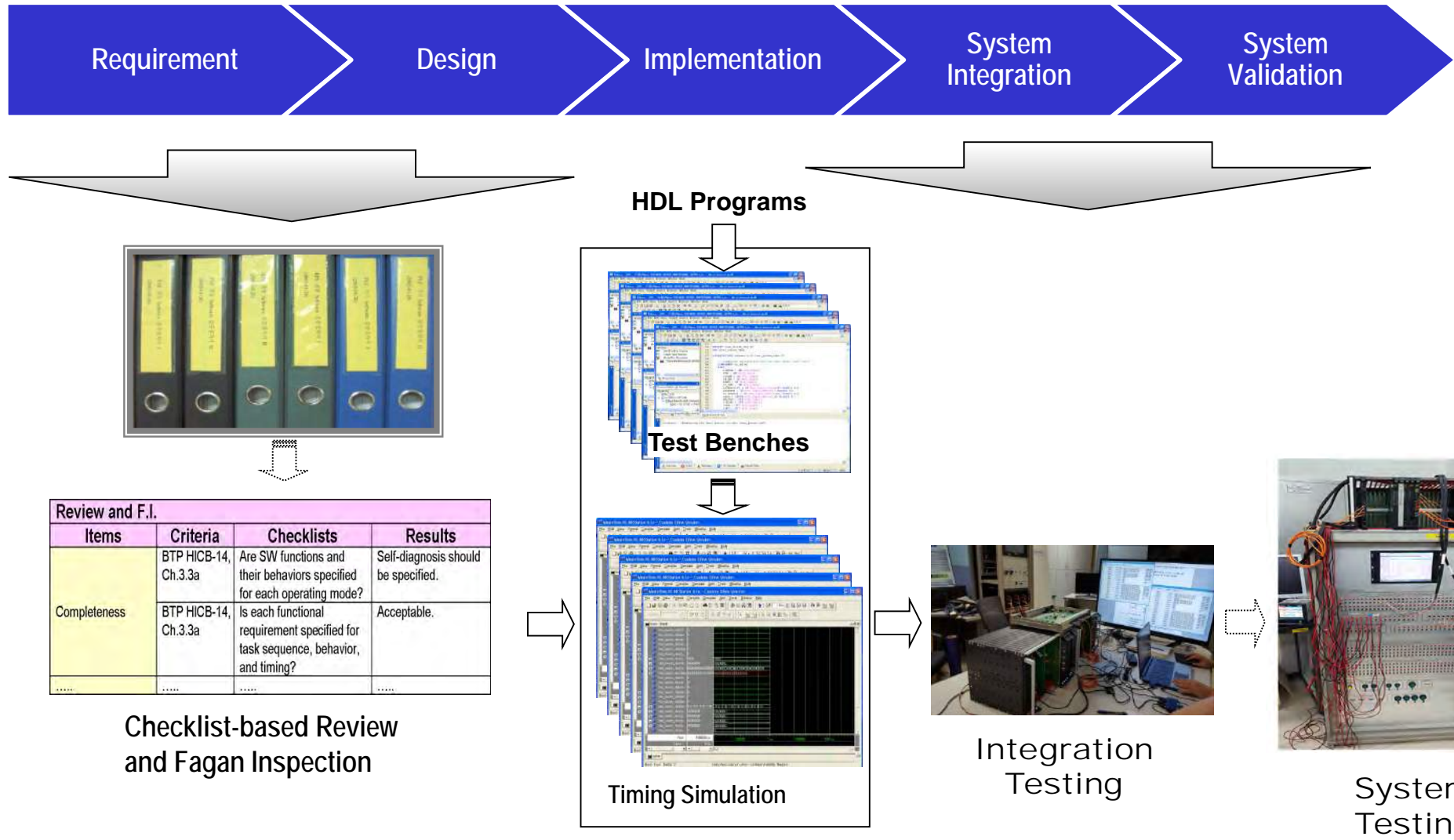


<Modified based on NUREG-0800>

# FLC- Code & Standard



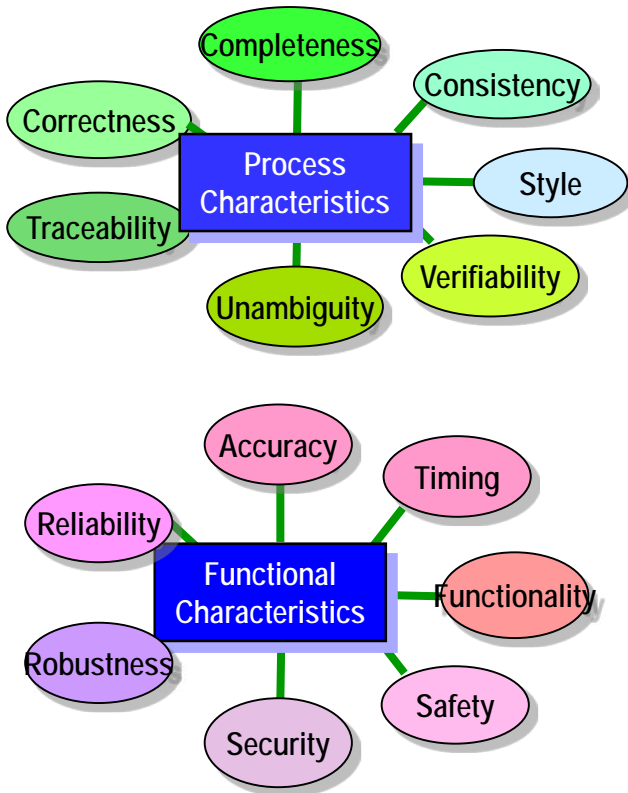
# FLC - Overall process



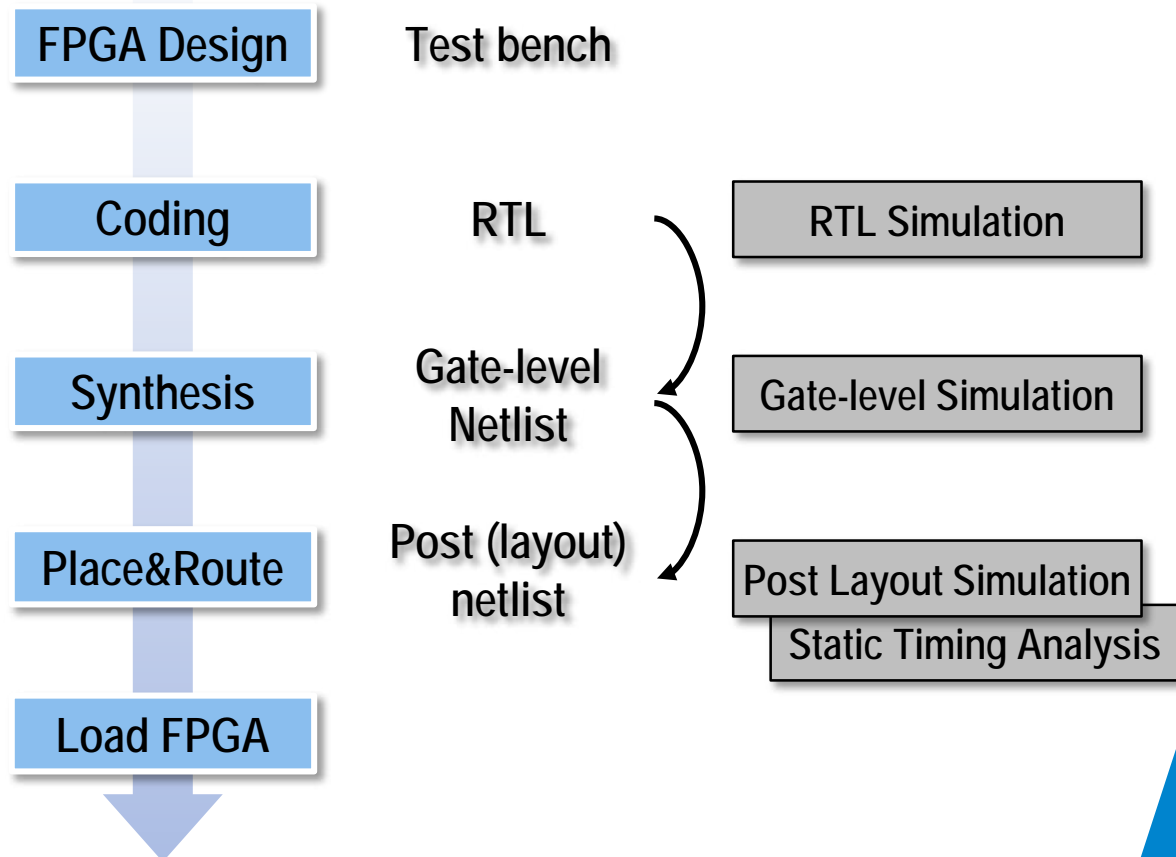
# V&V Activities for an FPGA – Review, CT



## SRS/SDS/Code Review



## CT(Component Test)



# V&V Activities for an FPGA - Integration Test

Concept

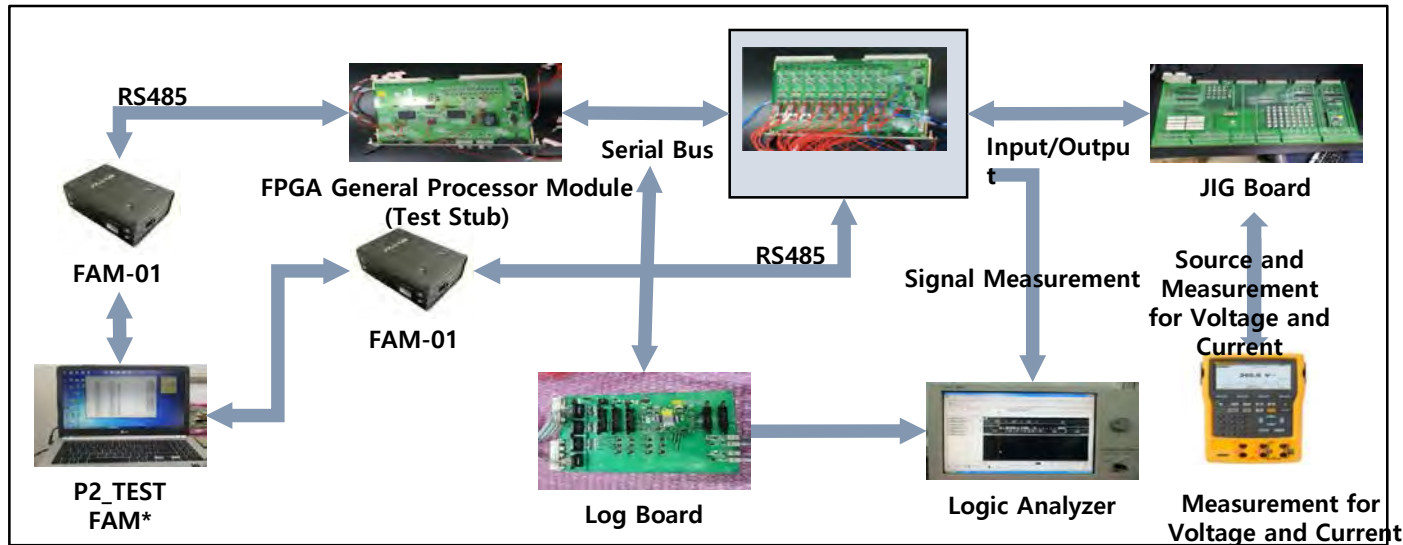
Requirement

Design

Implementation

Test

- Integration test is performed using hardware signal triggering and monitoring
- Test criteria : Logic requirement coverage
- Test environment : Jig board and signal jumpers for monitoring



## ◀ Test Environment for I/O Module Integration Test

### The Execution for I/O ► Module Integration Test



\* FAM(self-developed FPGA Adapter Module)

# V&V Activities for an FPGA - System Test

Concept

Requirement

Design

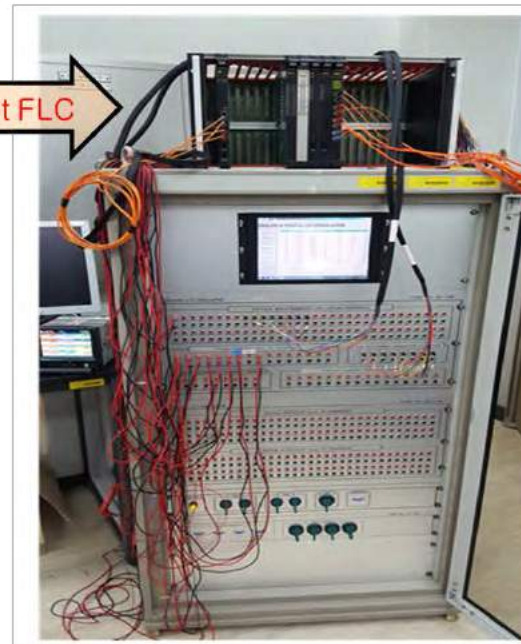
Implementation

Test

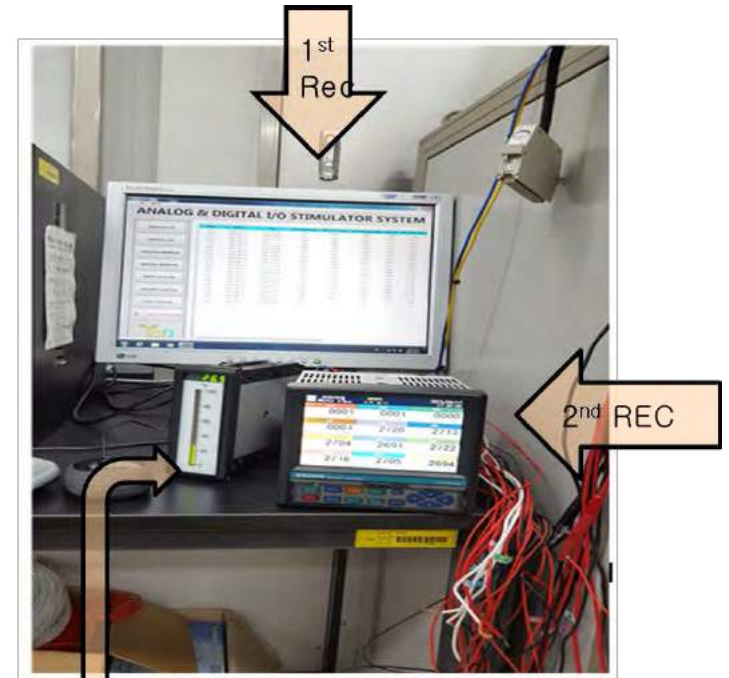
- System is validated by hardware-in-the-loop test
- Test criteria : system requirement coverage
- Test environment: Hardware-in-the-loop simulation environment

- System Test
  - Functional Test
  - Performance Test
  - Interface Test
  - Real-time Test
  - Fault Injection Test
  - Scenario based Test

Target FLC



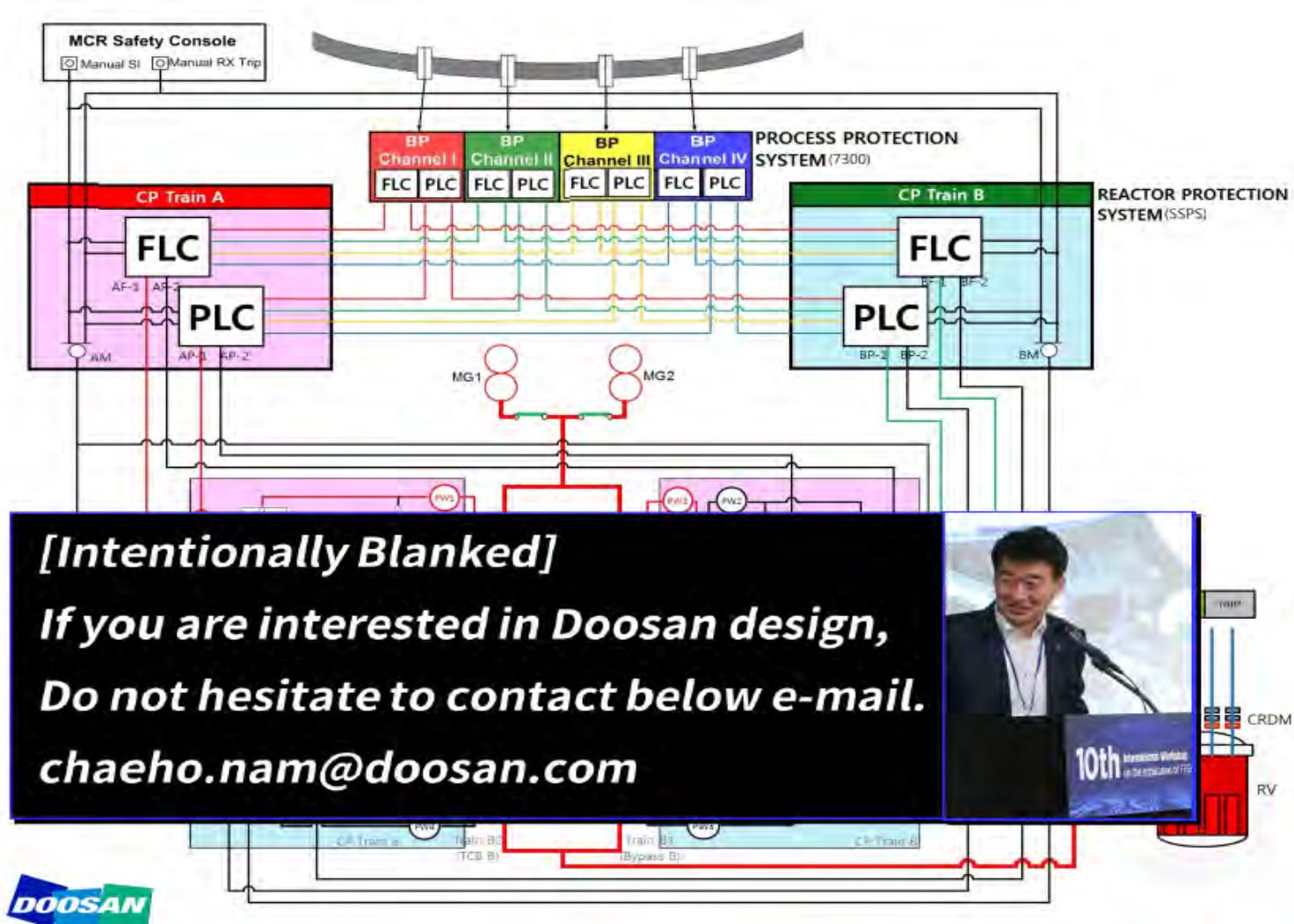
Automatic Signal Generator



Test Result

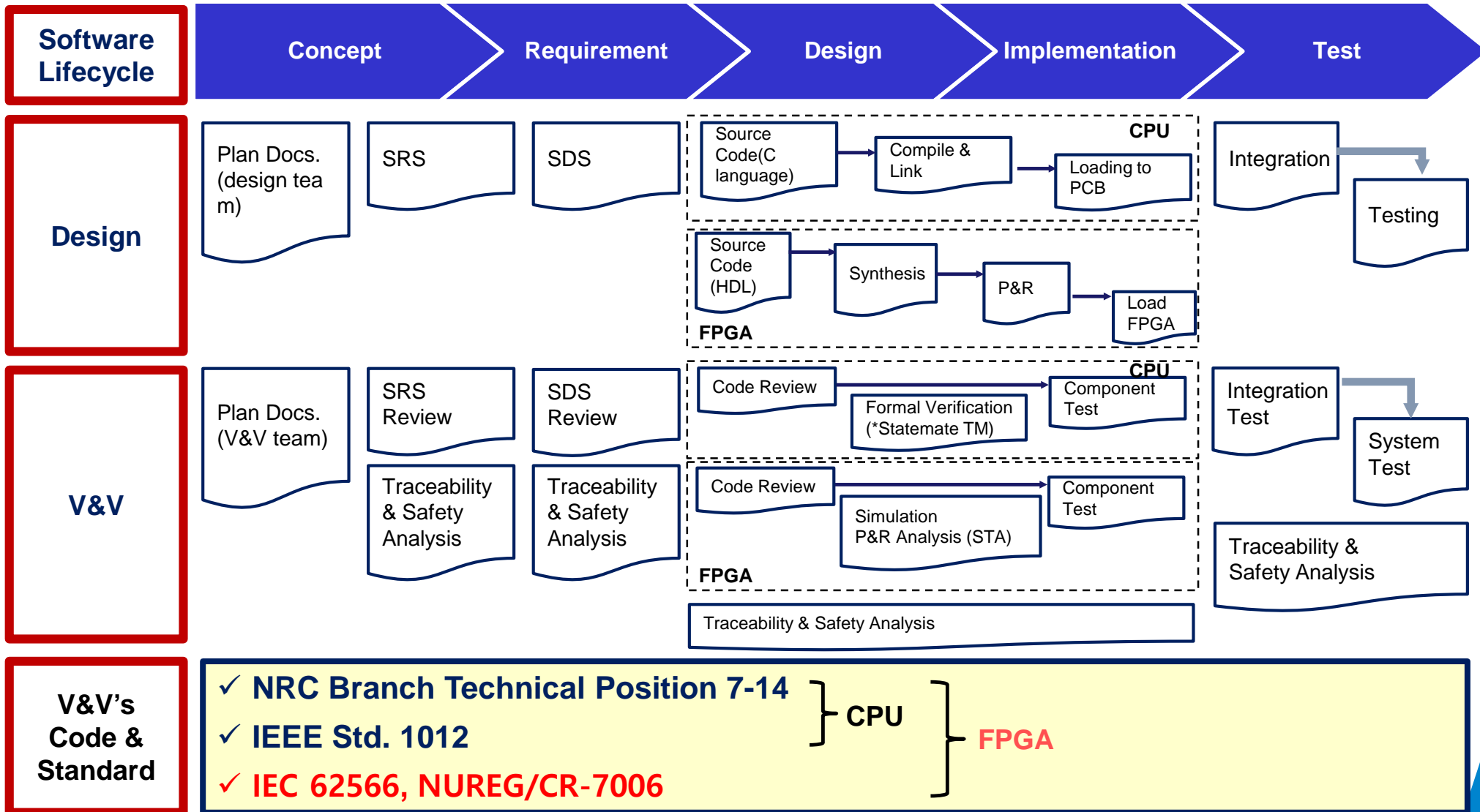
# FLC(FPGA Logic Controller) Application

- Reducing the CCF by using different platforms(PLC+FLC) for the protection system



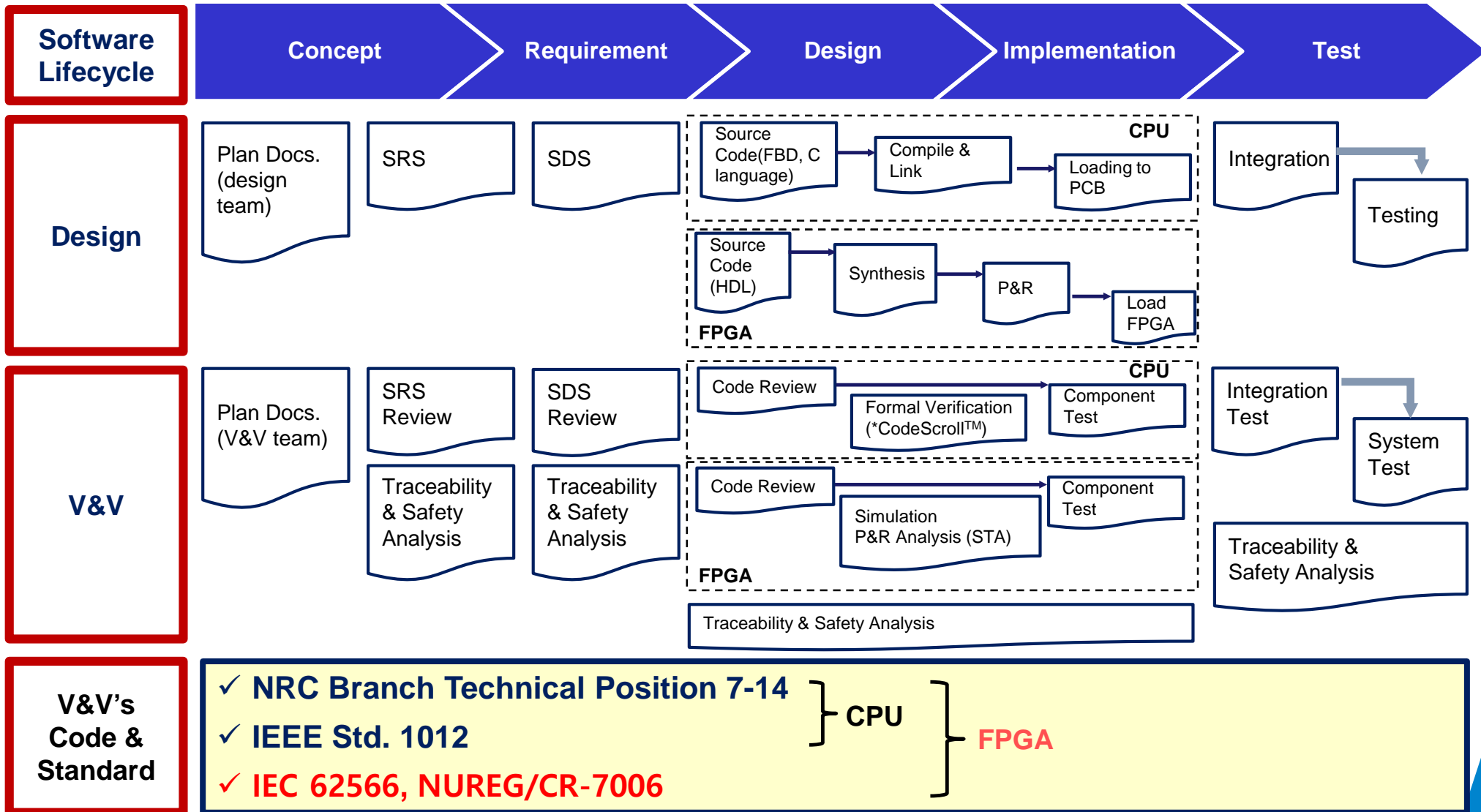


# Lifecycle comparison - Platform



\* Formal verification method by Statemate™ had been only used in initial R&D stage when developing PLC's operating system.

# Lifecycle comparison - Application



\* Formal verification method by CodeScroll™ is being used in program written in C language.

# Lifecycle comparison – IEEE 1012 vs IEC 62566

	IEEE 1012-2014	IEC 62566-2012
<b>Life cycle model</b>	<ul style="list-style-type: none"> <li>▪ This corresponds to the life-cycle defined in IEEE 1074-2006.</li> </ul>	<ul style="list-style-type: none"> <li>▪ This corresponds to the life-cycle defined in IEC 61513-2011, IEC 60880-2006.</li> </ul>
<b>Concept</b>	<ul style="list-style-type: none"> <li>▪ Current phase specifies the V &amp; V activities, required inputs and outputs.                             <ul style="list-style-type: none"> <li>✓ Requirements assignment analysis (correctness, accuracy, completeness)</li> <li>✓ Traceability analysis and risk analysis</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>▪ Current phase specifies the following requirements.                             <ul style="list-style-type: none"> <li>✓ V &amp; V plan shall be established by an independent verification team from the design team.</li> </ul> </li> </ul>
<b>Requirement</b>	<ul style="list-style-type: none"> <li>▪ Current phase specifies the V &amp; V activities, required inputs and outputs.                             <ul style="list-style-type: none"> <li>✓ Software requirements assessment, interface analysis(correctness, consistency, completeness, accuracy and testability)</li> <li>✓ Traceability analysis(correctness, consistency, completeness and completeness)</li> <li>✓ Risk analysis and prepare test plan</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>▪ Current phase specifies the following requirements.                             <ul style="list-style-type: none"> <li>✓ The requirements must be documented and verifiable so that the FPGA's electrical characteristics (frequency, fan-out and power-on profile), functional characteristics, deterministic characteristics, fault detection and fault tolerance characteristics must be created.</li> <li>✓ Completeness and consistency of the requirements shall be reviewed.</li> </ul> </li> </ul>
<b>Design &amp; Implementation</b>	<ul style="list-style-type: none"> <li>▪ Current phase specifies the V &amp; V activities, required inputs and outputs..                             <ul style="list-style-type: none"> <li>✓ Software specification evaluation, source code evaluation and interface analysis(correctness, consistency, completeness, accuracy, testability)</li> <li>✓ Traceability analysis(correctness, consistency, completeness)</li> <li>✓ Risk analysis</li> <li>✓ Prepare test procedures.</li> <li>✓ Perform the test.</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>▪ Current phase specifies the following requirements.                             <ul style="list-style-type: none"> <li>✓ HDL coding recommendations shall be developed so that there is no difference in behavior between functional simulation and post-synthesis simulation.</li> <li>✓ You should use a synchronous circuit.</li> <li>✓ Consider electrical characteristics of IC when power on / off</li> <li>✓ After P &amp; R, it is necessary to confirm that simulation is identical to RTL simulation by cycle-by cycle.</li> <li>✓ Static timing analysis shall be performed.</li> <li>✓ A test bench test shall be performed to simulate the required logical state.</li> <li>✓ The test shall establish test coverage criteria.</li> <li>✓ You shall review the completeness of the HDL specification(sensitivity list of process statement, completeness of case statement, etc.) through static verification activities.</li> </ul> </li> </ul>
<b>System Integration</b>	<ul style="list-style-type: none"> <li>▪ Current phase specifies the V &amp; V activities, required inputs and outputs.                             <ul style="list-style-type: none"> <li>✓ Prepare test procedures.</li> <li>✓ Perform the test.</li> <li>✓ Traceability analysis(correctness, completeness)</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>▪ Current phase specifies the following requirements.                             <ul style="list-style-type: none"> <li>✓ System integration procedure shall be established.</li> <li>✓ It shall include the interface and basic operating features of the FPGA as well as protocol, timing, and electrical characteristics.</li> </ul> </li> </ul>
<b>System Validation</b>	<ul style="list-style-type: none"> <li>✓ Risk analysis</li> </ul>	<ul style="list-style-type: none"> <li>▪ Current phase specifies the following requirements.                             <ul style="list-style-type: none"> <li>✓ System validation procedure shall be established.</li> <li>✓ System validation test is to verify that the FPGA can perform the required Category A functions through dynamic and static input signal.</li> </ul> </li> </ul>

# Feedback from Experience on CPU and FPGA - Regulator's perspective

	CPU	FPGA
<b>Application field</b>	<ul style="list-style-type: none"> <li>FPGAs and CPUs do not compete with each other. When you implement the system, you choose to use one suitable for FPGA or CPU according to the purpose of the system.</li> <li>CPUs are suitable for complex and logic-rich systems, while FPGAs are suitable for sequential logic.</li> <li>Many logic (functions) that the FPGA can not accommodate easily can be implemented in CPU.</li> </ul>	<ul style="list-style-type: none"> <li>FPGAs are concurrent (parallel) driven and suitable for timing-sensitive logic (However it needs careful analysis for complex logics)</li> <li>Because the logic is simple, verification is relatively easy because it does not use the operating system in the area where the function can be implemented by the FPGA.</li> <li>Applicable as means to satisfy diversity of CPU (considering SW CCF).</li> <li>Cyber security does not have much advantage over CPU.</li> </ul>
<b>Review Focus(Effort)</b>	<ul style="list-style-type: none"> <li>Checking the compatibility of CPU operating system</li> <li>In case of CPU, I&amp;C vendor trusts the timing of the logic circuit in the CPU because it is guaranteed in the proposed environmental conditions (voltage / temperature, etc.).</li> </ul>	<ul style="list-style-type: none"> <li>The timing of the logic circuit must be checked.</li> <li>Since the FPGA is developed directly to implement the logic circuit inside the FPGA, the timing satisfaction must be confirmed through simulation and testing.</li> </ul>
<b>Regulatory laws and standards</b>	<ul style="list-style-type: none"> <li>The requirements of IEEE 7-4.3.2, NUREG 0800 (SRP) BTP 7-14 apply basically as they are reviewed from a software development standpoint.</li> </ul>	<ul style="list-style-type: none"> <li>FPGAs are essentially hardware. However, the requirements of IEEE 7-4.3.2 and NUREG 0800 (SRP) BTP 7-14 are applied basically because they are regulated in terms of software development in that they use development languages and software tools.</li> <li>Some of the requirements of IEC 62566 apply (No US NRC regulatory guidelines for hardware characteristics of FPGAs).</li> </ul>
<b>Licensee Effort</b>	<ul style="list-style-type: none"> <li>CPUs and FPGAs implement logic in a fundamentally different way and thus have different verification methods (verification Effort).</li> <li>Licensing of which is easier than other can not be found.</li> </ul>	<ul style="list-style-type: none"> <li>Reduce the burden on operating system verification (even if FPGAs are added with timing verification)</li> </ul>

# Feedback from Experience on CPU and FPGA- Designer's perspective

	CPU	FPGA
<b>Platform development</b>	<ul style="list-style-type: none"> <li>Platform vendor should develop CPU operating system(OS) and application programming tool.</li> </ul>	<ul style="list-style-type: none"> <li>Platform vendor does not need OS and development tools.</li> </ul>
<b>Application program development</b>	<ul style="list-style-type: none"> <li>Platform vendor provides an application programming tool that supports industrial standard languages such as FBD (Function Block Diagram) (Platform vendor also performs verification of development tools directly)</li> <li>Program developers utilize graphical programming tools (pSet) to place Effort on the consistency of logic implementations.</li> <li>The program developer trusts the conversion process (Compile, Link) inside the programming tool. For example, compiler is dedicated or tested by platform vendor.</li> </ul>	<ul style="list-style-type: none"> <li>The program developer puts effort into the conversion process (synthesis and placement, timing, and physical characteristics of the FPGA) within the FPGA chip as well as the consistency of the logic specification.</li> <li>Compile Synthesis and system routing software are trusted.</li> </ul>
<b>Timing characteristic</b>	<ul style="list-style-type: none"> <li>Application response depends on scan time (input, logic execution, repetition cycle of output processing) and deterministic characteristic margin (load ratio) of OS. The deterministic characteristic of the platform depends largely on the determinism of the OS.</li> </ul>	<ul style="list-style-type: none"> <li>The response of the application depends on the timing characteristics between the scan time (input, logic execution, repetition period of the output processing) and the logic circuit determined after the physical placement of the FPGA chip.</li> </ul>
<b>Application program Size-up</b>	<ul style="list-style-type: none"> <li>Program size extension is determined by application program scan time adjustment in a range that does not exceed the maximum value of the program load ratio (platform determinism) and the input / output number does not affect the load rate.</li> </ul>	<ul style="list-style-type: none"> <li>Whether to extend the program depends on the physical resources and performance within the FPGA chip.</li> </ul>

# Feedback from Experience on CPU and FPGA- V&V perspective(1)

	CPU	FPGA
<b>V &amp; V Type (quantity) and difficulty of the document.</b>	<ul style="list-style-type: none"><li>▪ The type and quantity of documents are similar.</li></ul>	<ul style="list-style-type: none"><li>▪ The type and quantity of documents are similar.</li><li>• FPGA V&amp;V is more difficult.<ul style="list-style-type: none"><li>✓ In FPGA, it is difficult to distinguish between HW side and SW side in document specification.</li></ul></li></ul>
<b>V &amp; V difficulty (Cost) at application S / W development</b>		<ul style="list-style-type: none"><li>▪ FPGAs are more difficult.<ul style="list-style-type: none"><li>✓ For FPGA HDL logic, IEC 62566 and NUREG / CR-7006 require more equality testing between HDL, Synthesis, and P &amp; R outputs, so more cost input is required.</li><li>✓ In case of FPGA, test preparation and test execution by manual work such as writing test bench is relatively large.</li><li>✓ On board testing (testing after integrating FPGA into PCB circuit) requires expensive equipment such as Logic Analyzer, Oscilloscope, and Signal Generator, and it requires a lot of work to control / manipulate each device.</li></ul></li></ul>

# Feedback from Experience on CPU and FPGA- V&V perspective(2)

	CPU	FPGA
<b>Coding error</b>	<ul style="list-style-type: none"> <li>Many errors are found in the CPU, but not because of the fundamental difference, but because of the large size and high logical complexity of the CPU type software.</li> </ul>	<ul style="list-style-type: none"> <li>Errors are also related to the level of design / developer, so it is difficult to distinguish between the two.</li> <li>Experience has shown that FPGAs are challenging to test, but not at the level of statistical conclusions that more errors are found.</li> </ul>
<b>Calculation function</b>		<ul style="list-style-type: none"> <li>In FPGA, functional verification has to be verified several times through Simulation Test and On board testing.</li> </ul>
<b>Timing</b>		<ul style="list-style-type: none"> <li>Unlike CPU, the result of FPGA is hardware of logic circuit, so there is no big difference between Simulation operation and real time operation.</li> <li>It is important to verify the behavior of the processes running in parallel.</li> <li>On-board testing requires testing the FPGA pin-to-pin response time.</li> </ul>
<b>Accuracy</b>	<ul style="list-style-type: none"> <li>There is no difference between the two.</li> </ul>	
<b>Availability</b>	<ul style="list-style-type: none"> <li>There is no difference between the two.</li> </ul>	<ul style="list-style-type: none"> <li>FPGA is a little bit easier to evaluate(FPGA has more hardware feature)</li> </ul>
<b>Robustness, Fault tolerance</b>	<ul style="list-style-type: none"> <li>Both can be validated by Code Inspection, Exception Tests, and Tests that generate errors.</li> </ul>	<ul style="list-style-type: none"> <li>In the case of code inspection, if the HDL code of the FPGA is complex, it is difficult to analyze and a lot of workload is required.</li> </ul>

# Conclusion

- Under current experience and development environment, FPGA system is not understood as a competitor to CPU system but a cooperater, more important than a complementary system for diversity.
  - For simple logic applications, FPGA system can be more effectively used for safety function.
  - For complex logic applications, well verified CPU system is more effective for safety function.
  - Harmonized use of FPGA and CPU system can improve safety of NPP by almost perfect diversification.
- In the future, whether FPGA system can replace CPU system or not is not easily predictable.
- To develop a new IEEE code for FPGA in a short time looks not effective and can lead to a new complexity. Current best idea is the same as what has been done, proving total equivalence IEC 62566 and IEEE 1012 for specific FPGA applications as shown in the presentation practice.