U.S. DEPARTMENT OF ENERGY | Office of NUCLEAR ENERGY

Clean. Reliable. Nuclear.

# SymPLe: Design, Development and Results of Verifiable FPGA-based PLC Architecture for Safety Critical Nuclear Power Applications

**Advanced Sensors and Instrumentation Annual Webinar**

**October 31 – November 1, 2018**

Matt Gibson[1] and Carl Elks[2]

**(1) ERPI, (2) Virginia Commonwealth University**

# Project Overview

**Goal:** Develop alternate technologies to implement critical and nuclear safety control systems that ensure deterministic verifiability of system integrity.

**Objectives:**

- Understand the non-deterministic aspects of current architectural approaches.
- Development and implementation a hardware based sequencer architecture, called SymPLe 1131, to validate a hardware based configurable sequence architecture and demonstrate prototype application.
- Discover and understand validation and verification (V&V) methods that can ensure the integrity of hardware based sequencer.
- Develop and demonstrate a SymPLe based design and demonstrate a prototype application.
- Demonstrate a commercial grade dedication of these Hardware prototype applications to validate the effectiveness of the methods.

# The Team

- **Matt Gibson, Program PI, EPRI**
- **Dr. Carl Elks, Co-PI, Virginia Commonwealth University**
  - **Dr. Ashraf Tantawy, Research Professor, Virginia Commonwealth University**
  - **Rick Hite, Smitha Gautham, Chris Deloglos, Athira Jayakumar - Virginia Commonwealth University**
- **Jason Moore, MathWorks**
- **Andrew Nack, Paragon Inc.**

# Schedule and Milestones

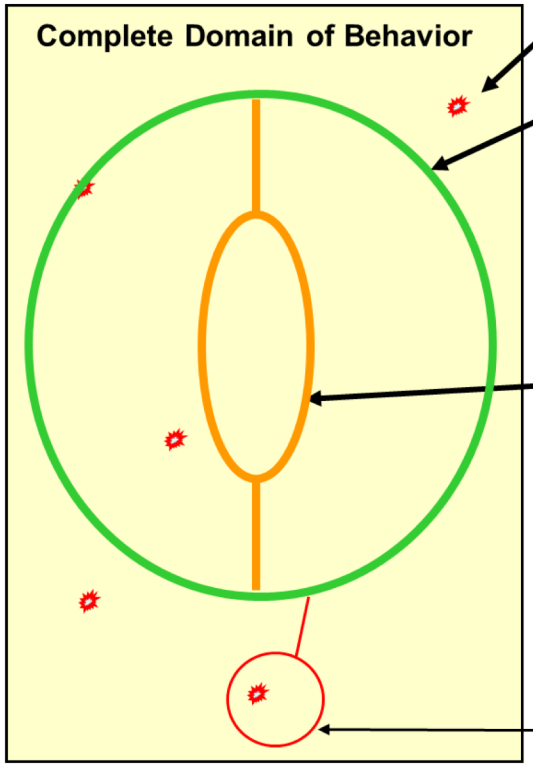
**Schedule:** Project is in third year with no cost extension

**Remaining Milestones:**

- 3/30/2019- Build and Demonstrate SymPLe1131 Prototype
- 3/30/2019-Perform and Document a Commercial Grade Dedication SymPLe 1131 Architecture
- 6/29/2019-Final Technical Report on Nuclear Qualification Demonstration of a Cost Effective Common Cause Failure Mitigation in Embedded Digital Devices-

# Software Based I&C

## Software Driven Architectures: Cyclic Behavior with Non-Deterministic Characteristics




May contain residual digital faults

***Path exercised continuously in normal situations***

**Influence factors** during continuous operation:

- normal process inputs (validated before use)
- short-term memory (as little as possible)
- clock interrupts (thorough verification)
- *(process-related interrupts: none)*
- *(resource management: static)*

***Path exercised in occasional but tested situations***

**Influence factors** that could disrupt cyclic behavior:

- initialization (only once)
- operator requests (single channel)
- hardware failures (single channel)
- exceptions (very simple)
- *(particular dates & times: avoided)*
- plant transients: affect all channels

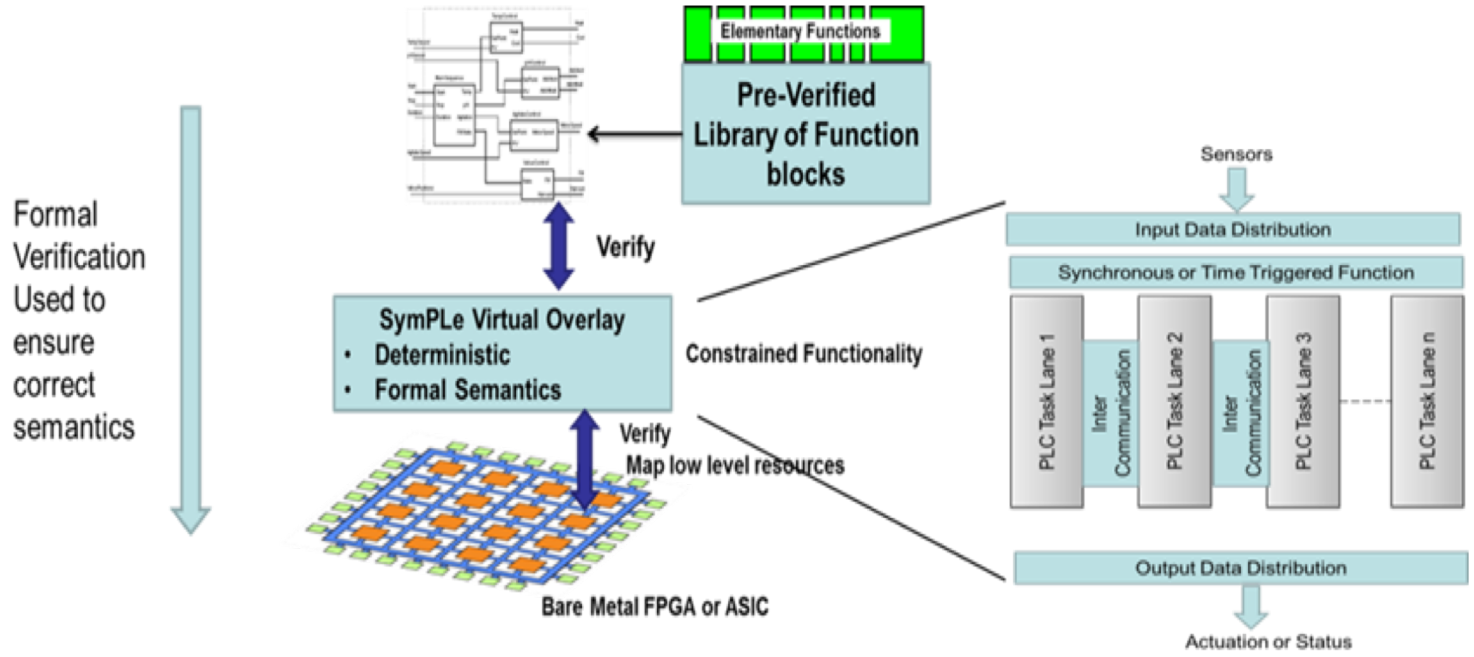
***Path exercised in unanticipated or untested situations***

**Can the System be *constrained* to well-understood and tested trajectories?**

# Technology Impact

- I&C systems in the context of nuclear power may not need to be derivatives of software intensive systems and by extension, not carrying the complexity associated with the SW intensive systems
  - This research eliminates large functional fault/attack surface that prevents unpredictable behavior when a SW based system exits the normal logic path due to software errors, misconfiguration, or malicious attack.
  - Maker Movement and low volume manufacturing of optimized components are now a reality and can be exploited to create components verified to function properly.
  - Digital components directly manufactured to 10CFR50 Appendix B are now within reach.
- Our approach called **SymPLe** is to rethink digital I&C from a perspective of three views: **Simplicity, Extensibility and Verifiability.**
- **Provides the Nuclear Industry a viable alternative to software based systems. This supports the DOE-NE mission to sustain and enable commercial nuclear power production.**

# SymPLe Concept




- **SymPLe is a *virtual machine or overlay***
  - *SymPLe* is a architectural viewpoint that seeks to maximize reasoning, transparency and safety evidence while avoiding unnecessary complexity.
  - *Engineer Accessible*: By adopting overlay architecture, we hope to make SymPLe extensible like a CPU based architecture – function blocks are the execution functions.
  - SymPLe is limited in what it can do – it trades computational power for verifiability.

# SymPLe Architectural Concepts

- Constrain design to favor verifiable execution behavior
- Constrain program composability rules to favor testing and comprehension
- Design for predictability
  - Well understood behaviors
  - Well formed semantics – no side effects
  - *Engineer Accessible*: SymPLe is explicitly specified in a manner (e.g., language; structure) that is comprehensible to the community of its users and reviewers.
  - Reusable and verified Hardware Function Blocks

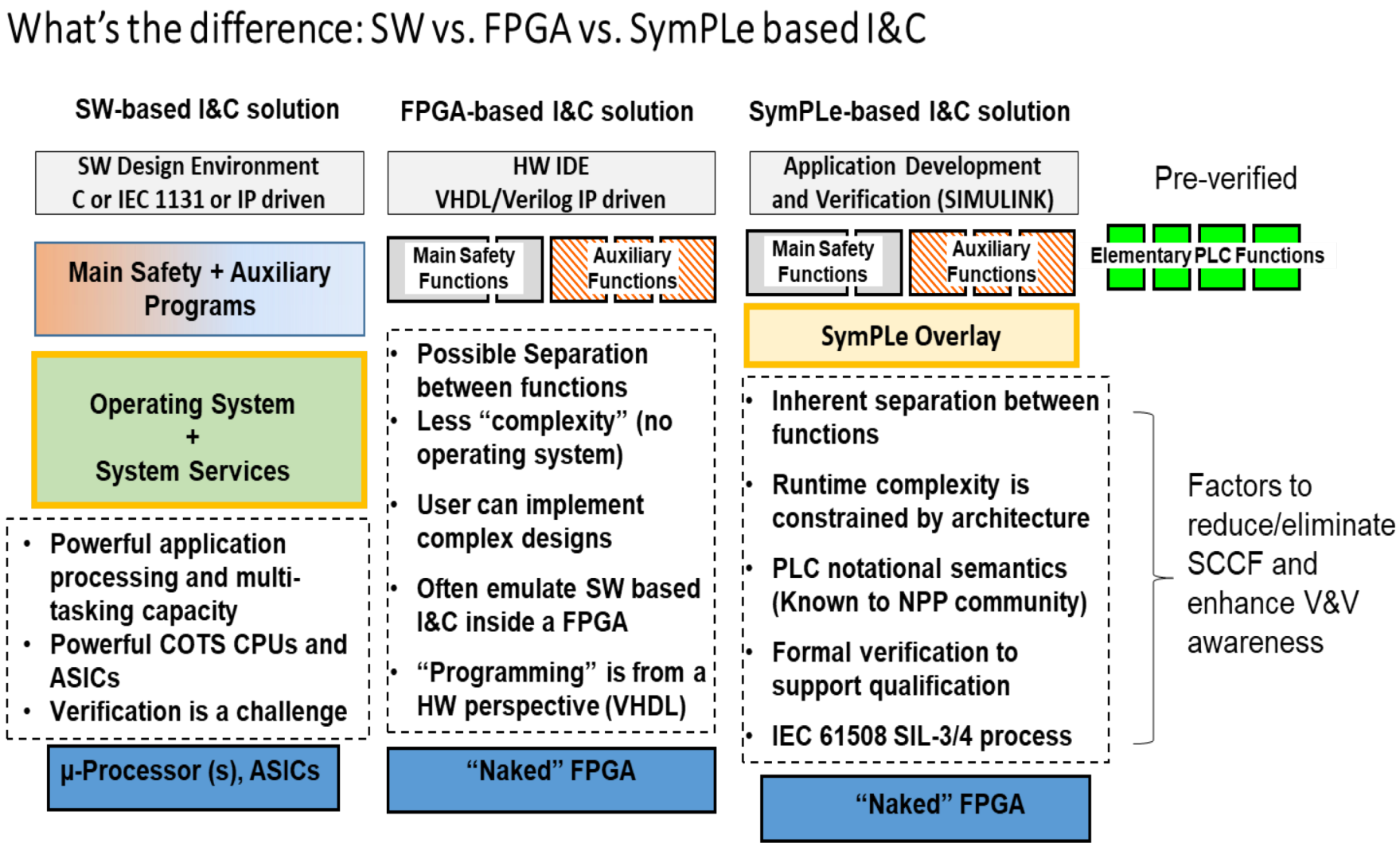
# Basic Tenants of SymPLe

- *Composability* - The behavior of "composed" element is a composition of the behaviors of its constituent elements, with well-defined, unambiguous rules of composition.
  - –Interfaces of elements are unambiguously specified, including behavior.
  - –Interactions across elements occur only through their specified interfaces
- *Orthogonal* - The system is modularized using principles of information hiding and separation of concerns, considering orthogonality[1] of functions and data.
  - –Only required interactions are allowed. The architecture precludes unwanted interactions and unwanted, unknown hidden coupling or dependencies.
  - –Each element (e.g., a FB unit) is internally well-architected and relatively simple.
- *Determinism* - The system is architected (satisfying conditions above) to be predictable and synchronous in it's execution behavior.

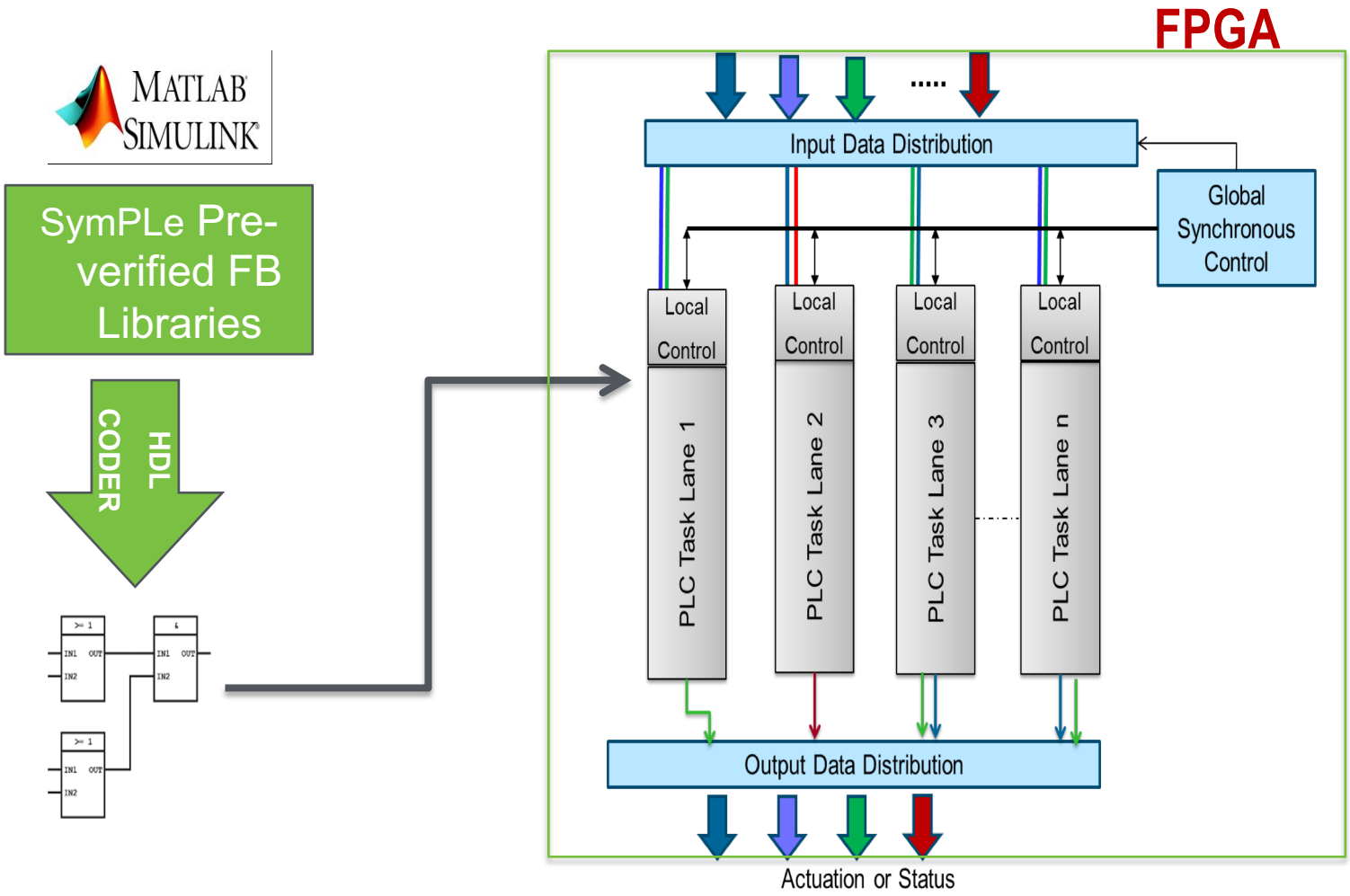
**These Tenants along with a formal design and verification methodology allows any CCFs to be identified before deployment, and enhances the ability to reason about system during qualification.**

1 = a relatively small set of primitive constructs can be combined in a relatively small number of ways to build the control and data structures of program behavior

# Comparative View of I&C "Stacks"



# High Level Architecture Model of SymPLe



# SymPLe Function Blocks

**Most I&C applications in NPP are not computationally challenging**

**Elementary FBs**

| Instruction | Description |
|---|---|
| AND, OR, NOT, XOR, NAND, NOR | Logical Operators |
| AND, OR, NOT, XOR, NAND, NOR | Bitwise Logical Operators |
| MAX, MIN, MUX | Selection Operators |
| GT, GE, EQ, LT, LE, NE | Comparison Operators |
| ADD, SUB, MUL, DIV | Arithmetic Operators |
| SLL, SLR | Bit-shift Operators |
| MOVE | System Operators |

**Built-up FBs**

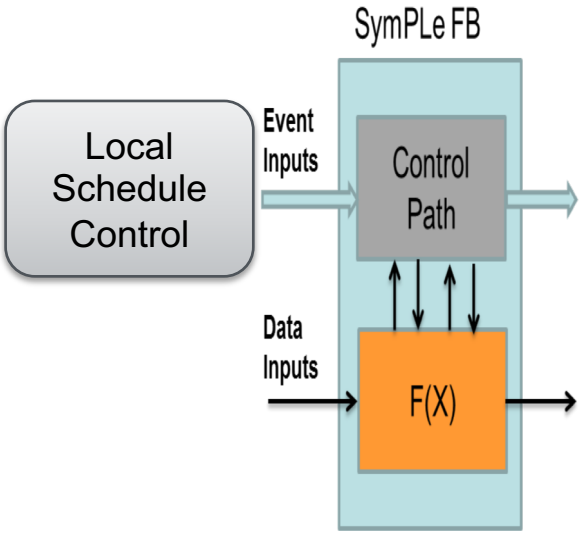
| Instruction | Description |
|---|---|
| PID | Control operator |
| TON, TOF | Timer operator |
| FXTOI, ITOFX | Data conversion operator |
| AVOTE | Analog voting |
| BVOTE | Digital voting |
| MEM | Memory operator |
| LOG | Logging operator |

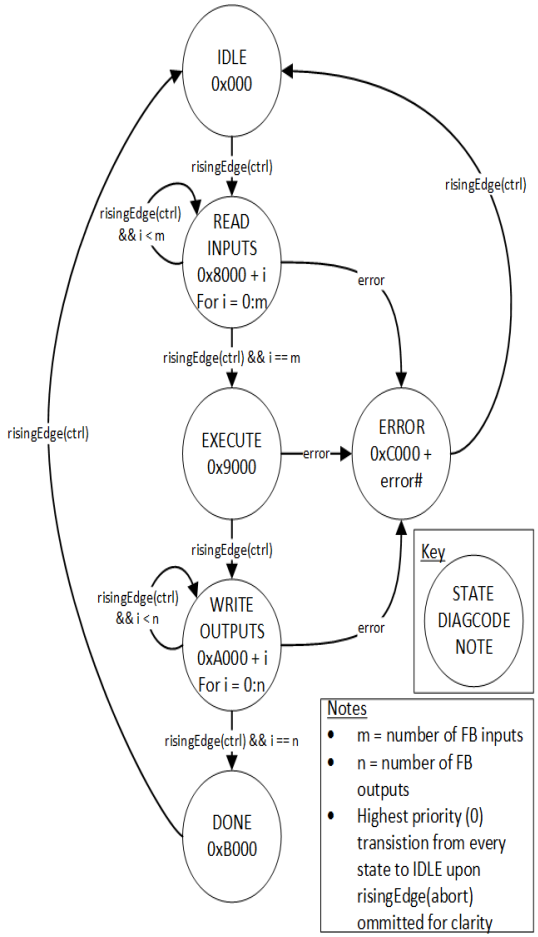
**All Function blocks V1.0 were formally proven. Working on V2.0**

# Function Block Architecture

Common architecture for all SymPLe function blocks

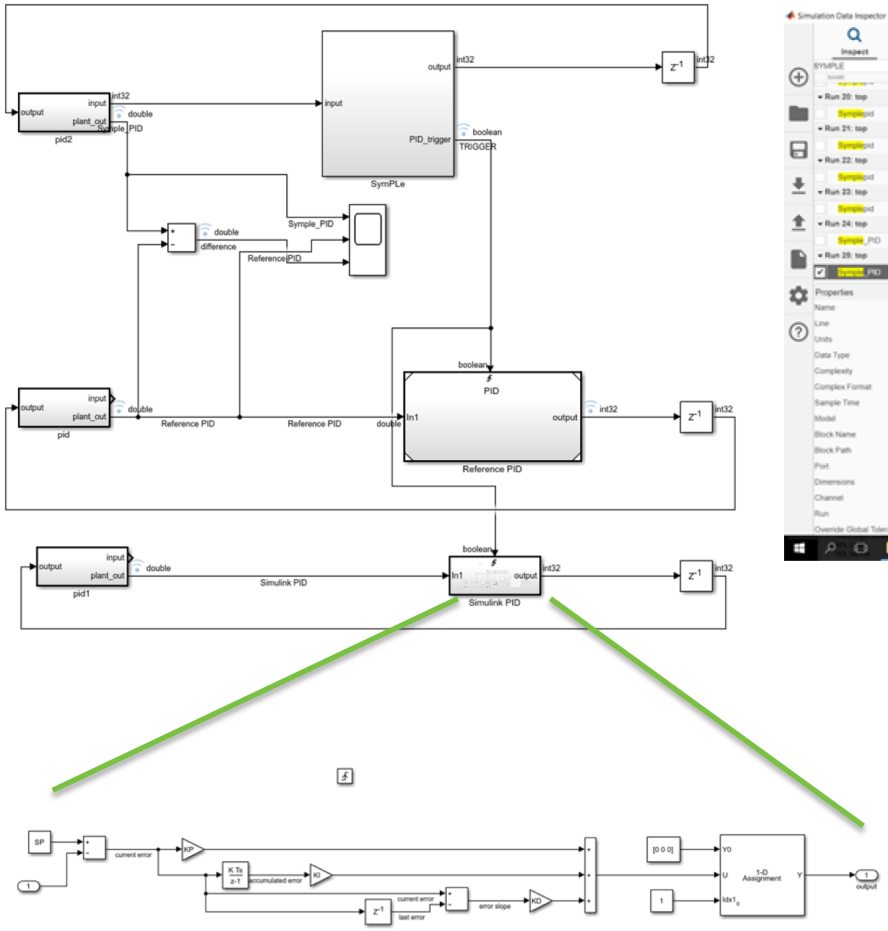
- Inspired by IEC 61499
- Deterministic, synchronous behavior.
- Separation of control and dataflow with clear and defined interconnections
- Formal semantics

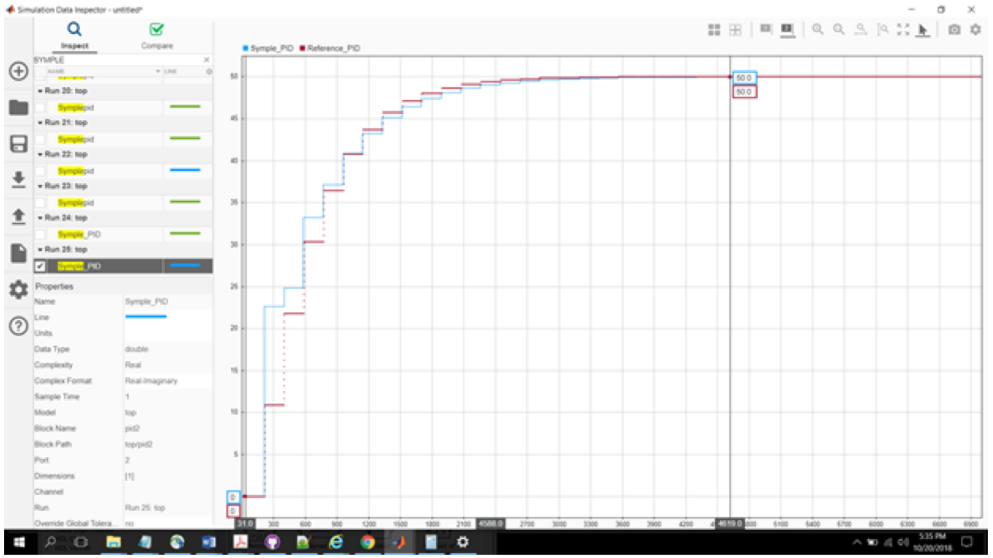


# Example: SymPLe PID Function Block

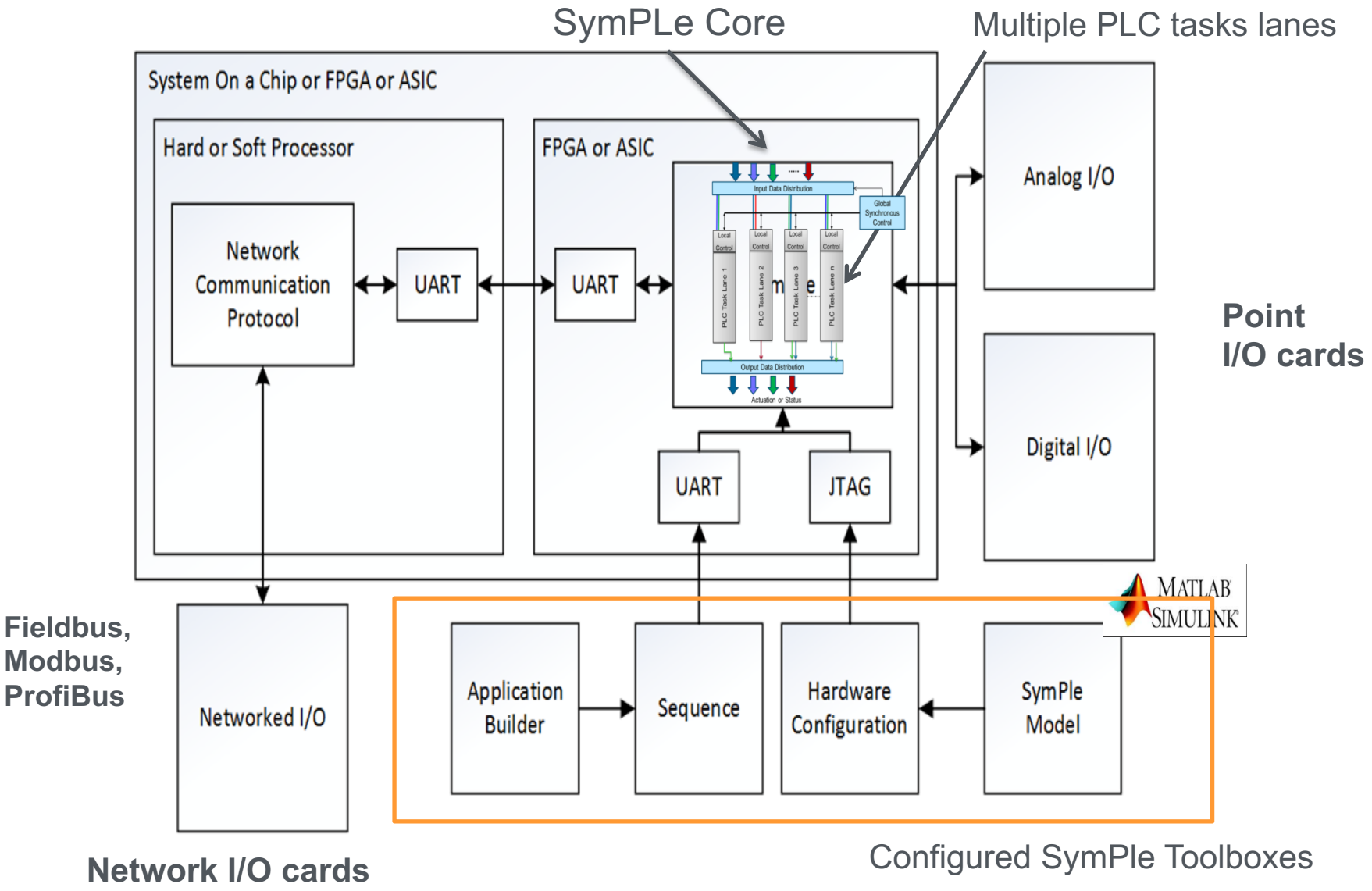
Symple PID Testing Test Model – Compare SymPLe PID to Simulink PID block under a variety of test conditions.



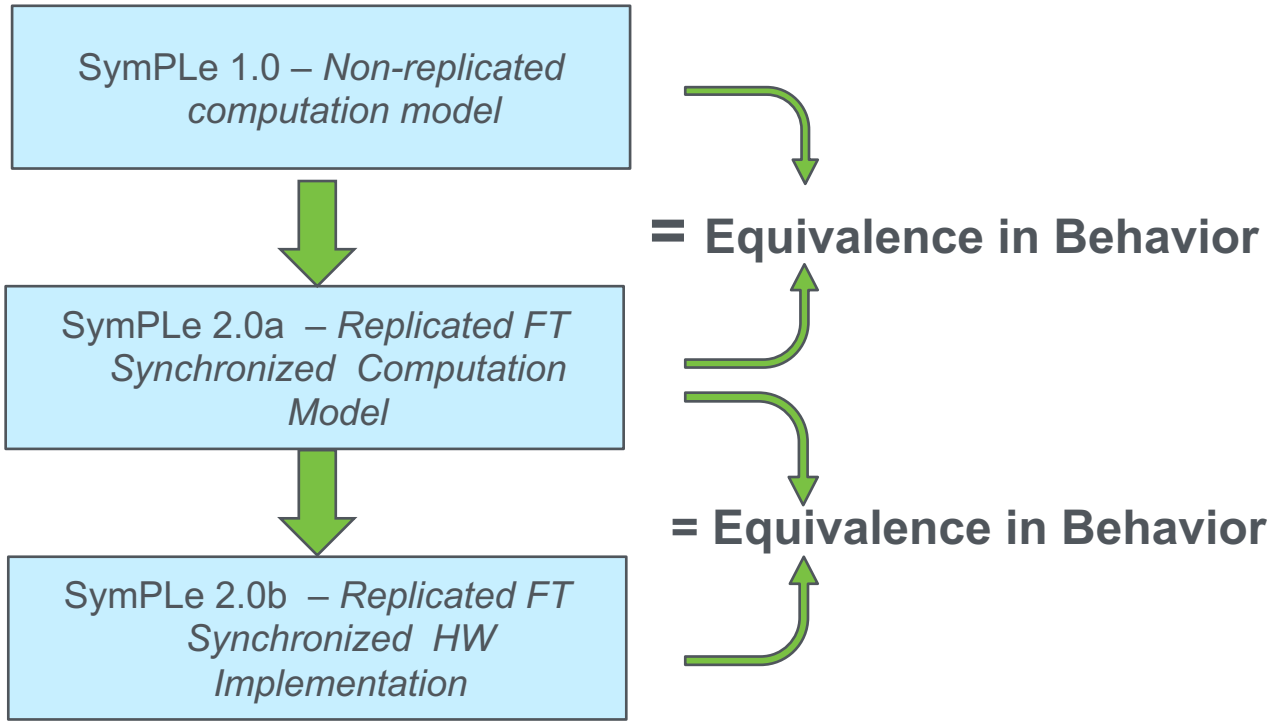
**SymPLe PID – Comparable to the Simulink code based PID. Some performance issues related to fixed point arithmetic of SymPLe. This is being fined tuned. PID is the most complicated block in SymPLe library.**

# Complete SymPLe System



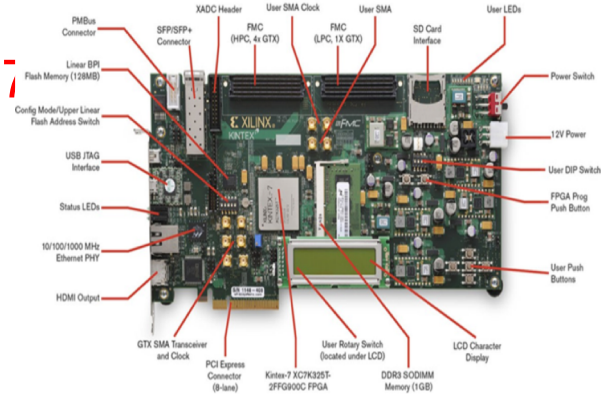
# SymPLe 2.0 – A Reliable Computing Platform

- In the design of SymPLe 2.0 we specified the FT design as a *transformation process* from a non-replicated computation machine V 1.0 to a replicated fault tolerant machine.
- SymPLe 2.0 has all of the operational semantics of SymPLe 1.0 , but it has new fault tolerance properties.



# SymPLe Initial Implementation – Xilinix FPGA

- Initial design was implemented on Xilinx KINTEX 7




MathWorks Simulink Toolboxes

FPGA (Xilinix) Tool and Mentor Graphics Synthesis

SymPLe Simulink Model → (Sim File) → HDL Coder → (HDL Code) → FPGA Synthesis → (Bit File) → 


Constrained Rules

Constrained and Verified Design

Equivalence checking

Constrained and Verifiable Implementation

# SymPLe Industrial Implementation

- Moving Beyond Xilinx Kintex 7 development board – we chose to implement SymPLe on a industrial controller to explore issues with portability, I/O interfacing, behavior in a real world platform.

- We chose the National Instruments cRIO Industrial controller with modular I/O capability.

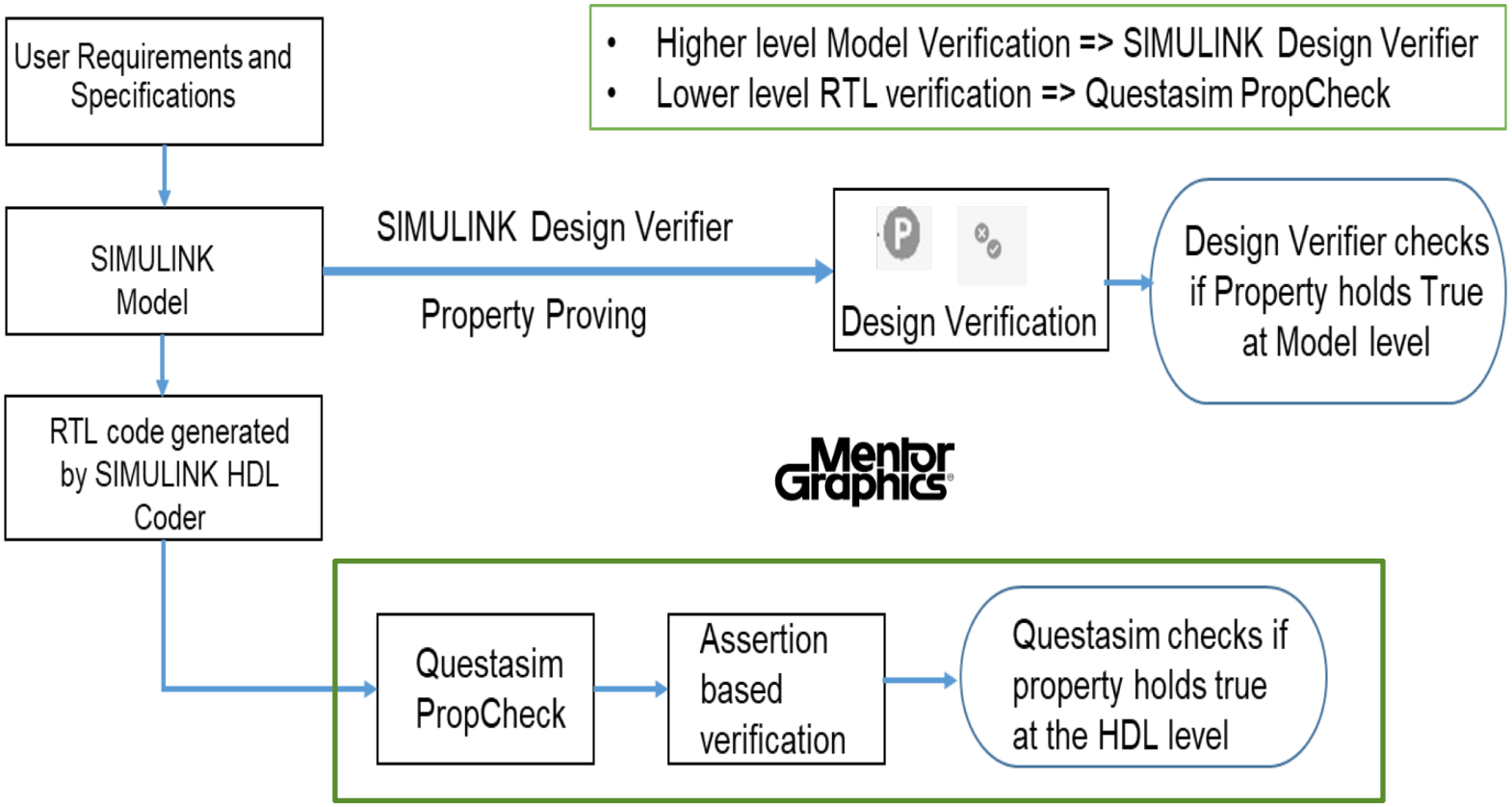
- ➢Built-in FPGA to enable the implementation of SymPLe architecture using HDL code.

  ➢Flexibility in I/O programming to interface with SymPLe.

# VCU Hardware In the Loop (HIL) Implementation

- Initial HIL was setup for the Emergency Diesel Generator (EDG) application.
- For GPWR HIL, use PID in feed water loop.
- A simple user interface was built using National Instruments LabVIEW software to manipulate EDG inputs and monitor the outputs.
- SymPLe architecture, implementing EDG logic, was downloaded to National Instruments cRIO controller FPGA.
- Using modular design, a SymPLe subsystem block was developed in LabVIEW, where it could be reused in other test programs.

# End-to-End Property Verification using Simulink DV and MENTOR GRAPHICS

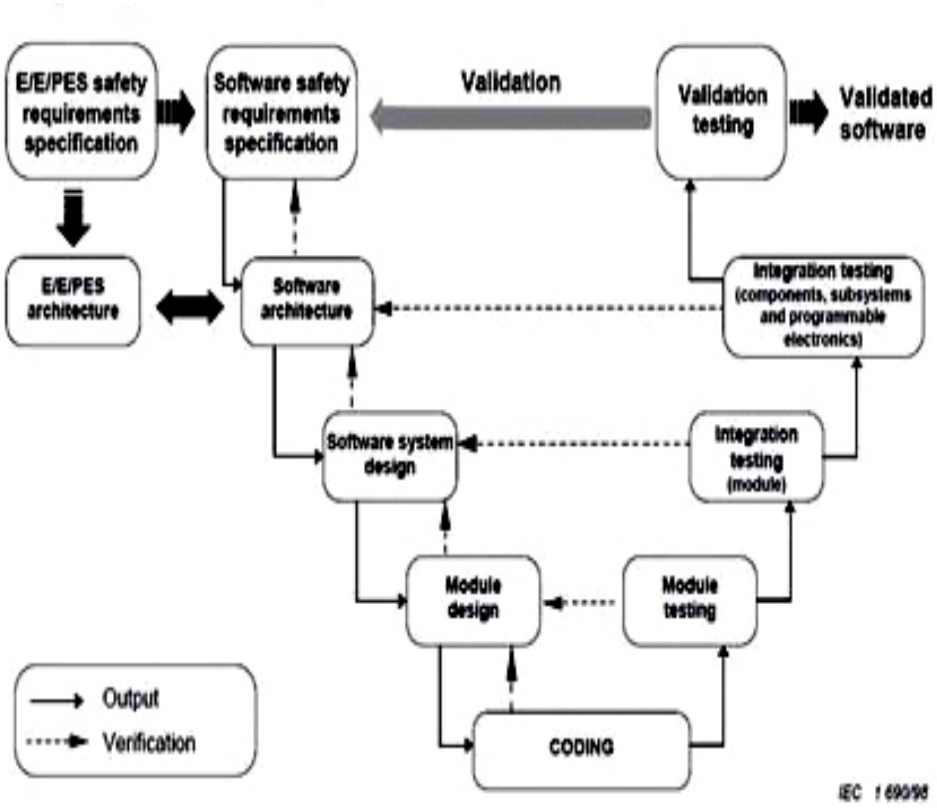



**Verifies that actual SymPLe HW has inherited the proven property of models**

# Commercial Grade Dedication Exercise

## To Stress the SymPLe concept – It is undergoing a limited and focused Commercial Grade Dedication(CGD) based on IEC 61508

- CGD will focus mostly section 3 of IEC 61508- Software Process
  - SymPLe was designed, developed, verified, and implemented with a Model-based design process
  - Our functional models of SymPLe are “system” oriented with no relation to HW or SW, but our process ultimately produces HDL code
  - Paragon Tech is performing “Mock” CGD.
- The Section 3 “V” model along with the corresponding annexes are closest to existing CGD activities.
- To our knowledge this is one of the first “model based design” efforts with respect to a Nuclear I&C to be CGD reviewed



# External Feedback

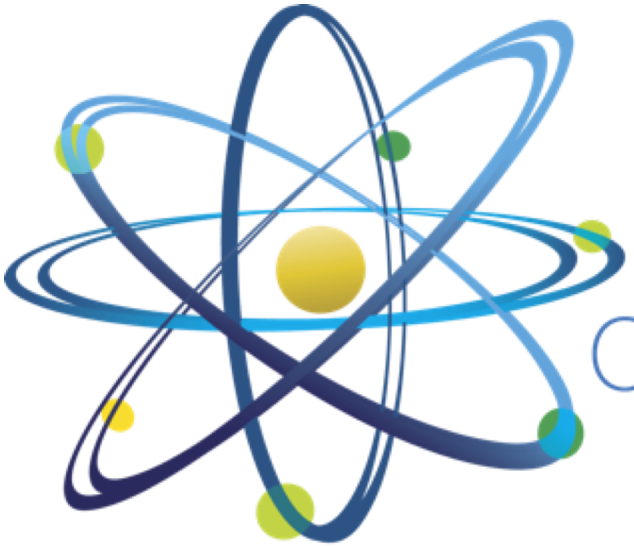
- SymPLe has been presented at a number of conferences, NRC meetings, workshops, and other venues.
- At present we have substantial interest from NuScale, OneSpin, Rock Creek, and Mathworks.
- We presented two talks at the 11th annual FPGAs in Nuclear Power Workshop in Dallas in mid-October.
  - Received very positive reviews from various I&C stakeholders in the Nuclear Power community.
  - One very positive pronouncement from NuScale, Rock Creek, and OneSpin was that our model based design approach was seen as the future for digital I&C.
  - We are scheduling follow up meetings to discuss future collaborations.

# 2018/2019 planned activities

- Conduct IEC 61508 (SIL 3)Commercial Grade Dedication (CGD) Exercise with Paragon Energy Solutions Nuclear and EPRI.
- Use Model Based Tools to provide CGD data to Paragon for CGD review.
- Complete GPWR Hardware in the loop by mid November.
- Continue refining SymPLe 2.0 architecture, complete critical review of Property Proving with MathWorks experts
- More HW testing, testing, collect data…….
- Write papers
- Initiate collaboration with interested parties.

# 2018 Conclusions and Take Away

- Designed and realized a practical verifiable  PLC Instantiable architecture that addresses CGD and SCCF via:
  - Constrained architecture operations
  - Formal deterministic FB semantics
  - Extensive use of model based design and analysis
  - Complementary Formal verification and Testing
- Tested Hardware in the Loop in the lab, working toward HIL for GSE GPWR.
- Demonstrated  SymPLe concept with Basic Emergency Diesel Generator (EDG) start controller, PID loops, etc…
- Developed design, models, and beta versions for ultra safe versions of SymPLe (Safety Integrity Level 3/4)
- Presented one paper at ANS UWC 2018, papers in preparation for IEEE Journal on Control Systems Technology and IEEE Dependable and Secure Systems and Network.



Clean. **Reliable. Nuclear.**

Contact Carl Elks, crelks@vcu.edu or Matt Gibson, mgibson@epri.com for further information