Semi-formal Reformulation of Requirements for Formal Property Verification

Katharina Ceesay-Seitz, CERN

Hamza Boukabache, CERN

Daniel Perrin, CERN

accellera SYSTEMS INITIATIVE

2019 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION EUROPE

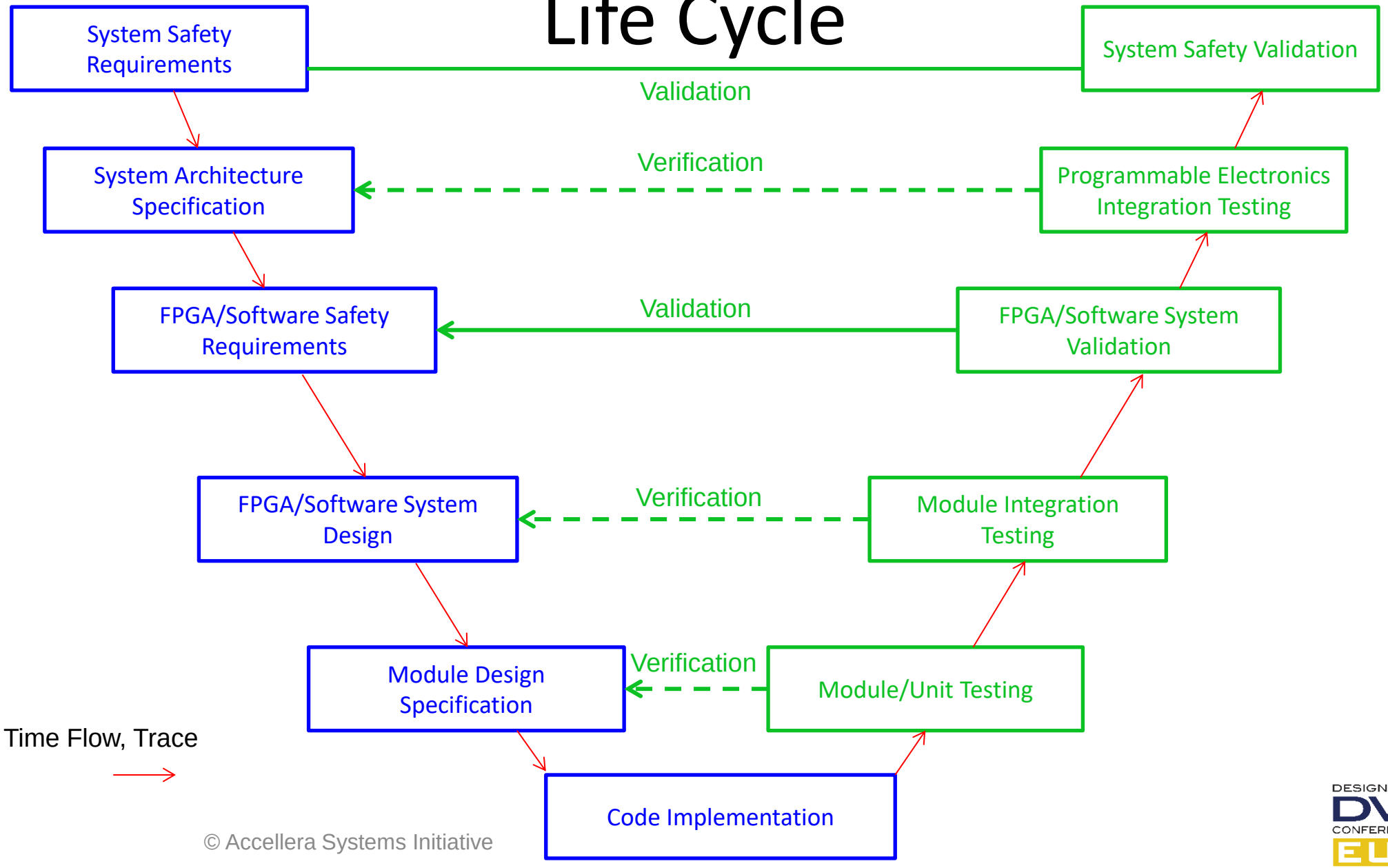
Agenda

- Motivation & Introduction
- Natural Language Properties (NLPs)
- Examples
- Results
- Conclusion

Motivating Example

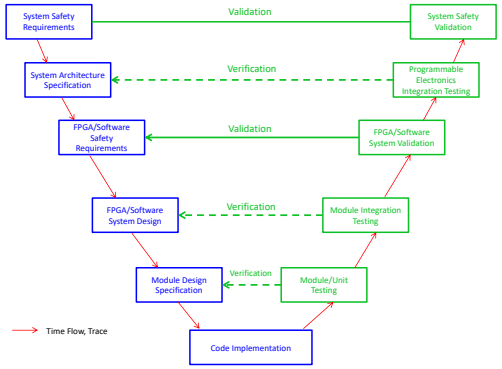
- 1999: Loss of NASA's Mars Climate Orbiter space craft
  - Onboard software and ground software used different measurement units
- 2018 + 2019: Crash of 2 Boeing 737 MAX 8 planes
  - Multiple reasons (sensor failure, lack of pilot training, safety features as add-ons to purchase, …)
- In both cases:
  - Lack of communication
  - Unvalidated assumptions

Life Cycle



Motivation

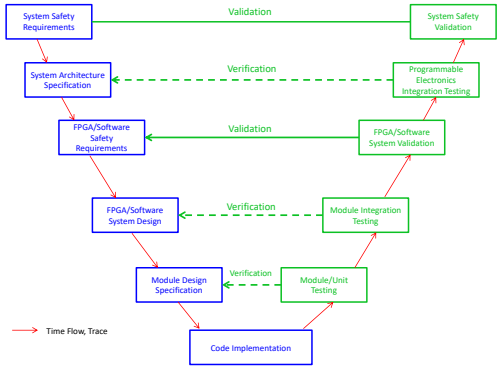
- Design specification is a refinement of requirements specification
  - Different levels of abstraction
  - Often written by different persons
  - Ambiguities → misinterpretations



- Misunderstandings enter the design specification
  - Can have life threatening consequences (safety)
  - Can cause unnecessary iterations (expensive)

Motivation & Introduction

- Validation of the design (specification) necessary
  - Proposed technique: Natural Language Properties (NLPs)
  - Validating the design specification through verification



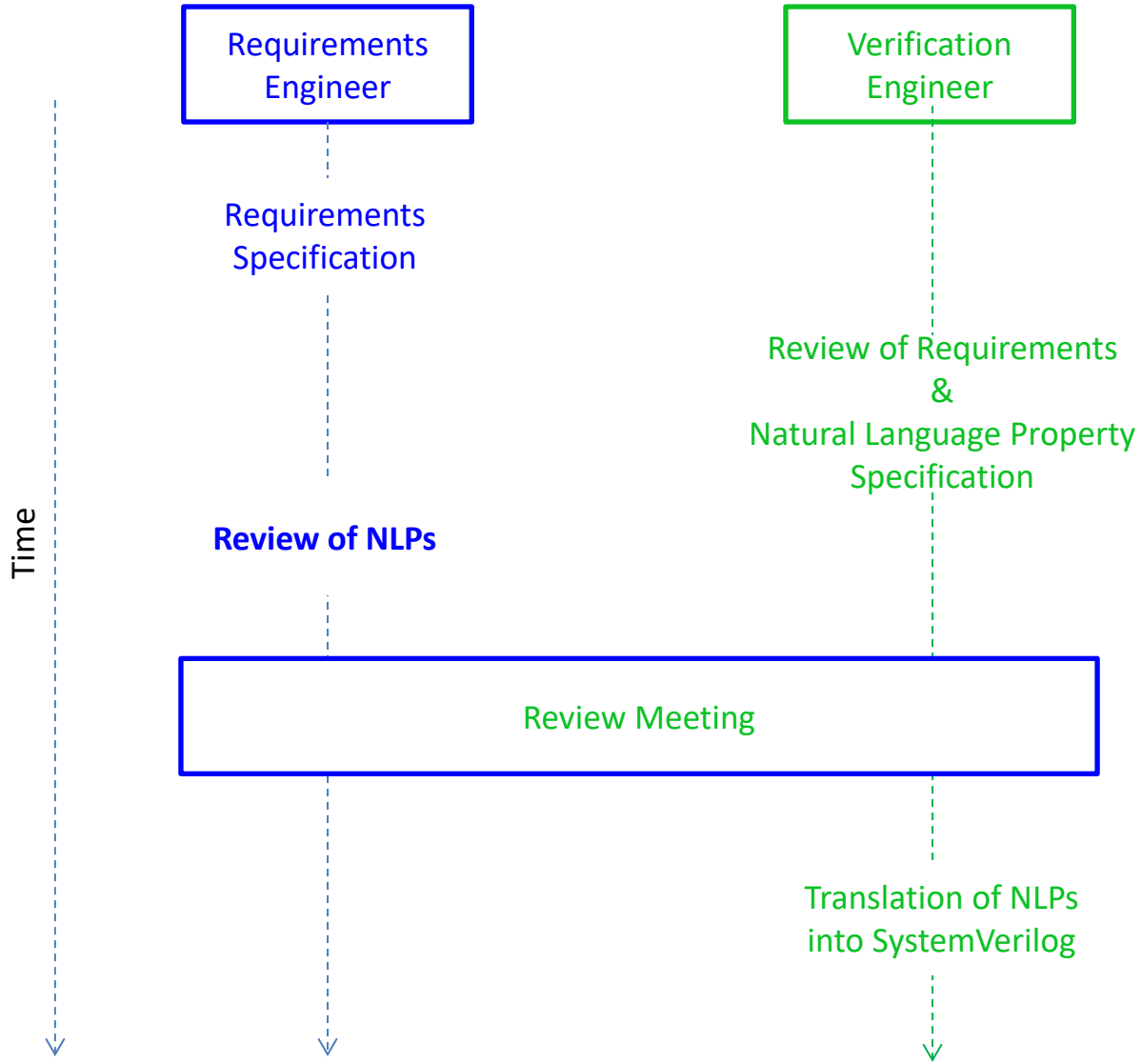
- Formal Property Verification
  - Exhaustive proof of formal properties
  - Bounded model checking
  - Property languages: SystemVerilog, PSL, …

Natural Language Properties (NLPs)

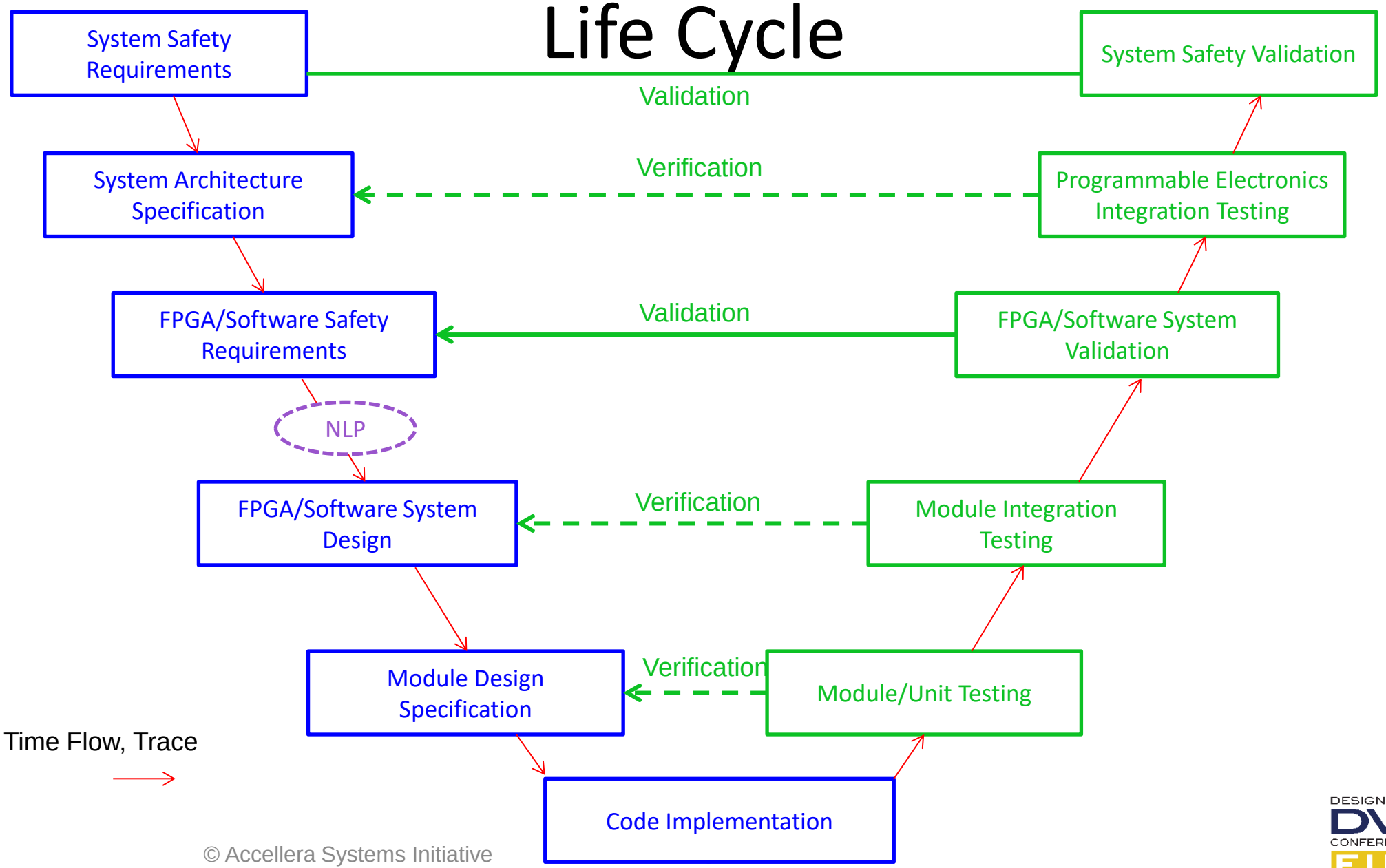
- Formal properties expressed in natural language
- Defined **“n:1”** mapping from **NLP** snippets **to SystemVerilog**

- **Review of NLPs** (and by that the formal properties) **by requirements engineer**
  - **Validates the verification**
  - Implicitly validates the specification when verification passes

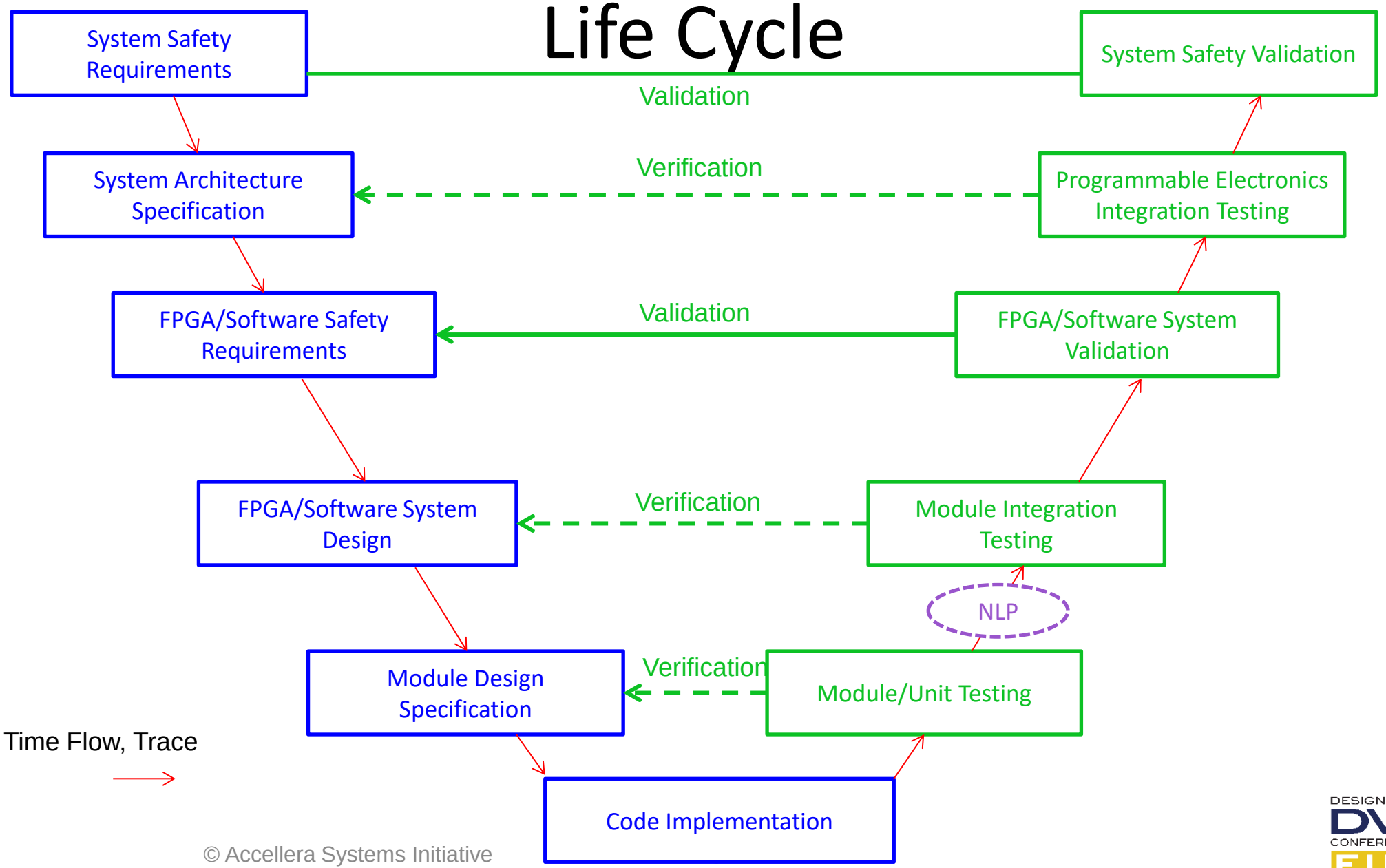
Workflow

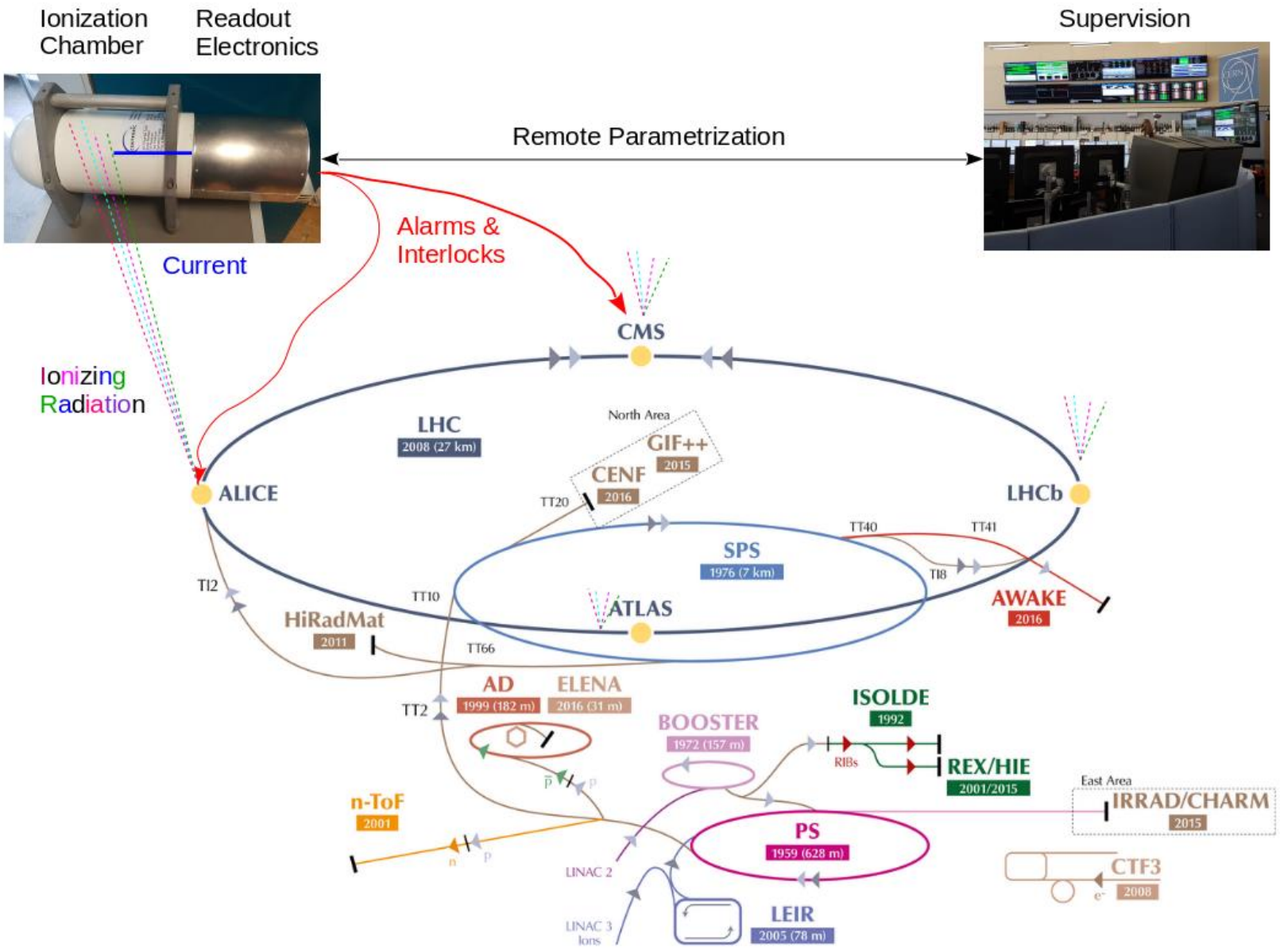

Requirements Engineer
Verification Engineer
Requirements Specification
Review of Requirements & Natural Language Property Specification
Time
Review of NLPs
Review Meeting
Translation of NLPs into SystemVerilog

Life Cycle



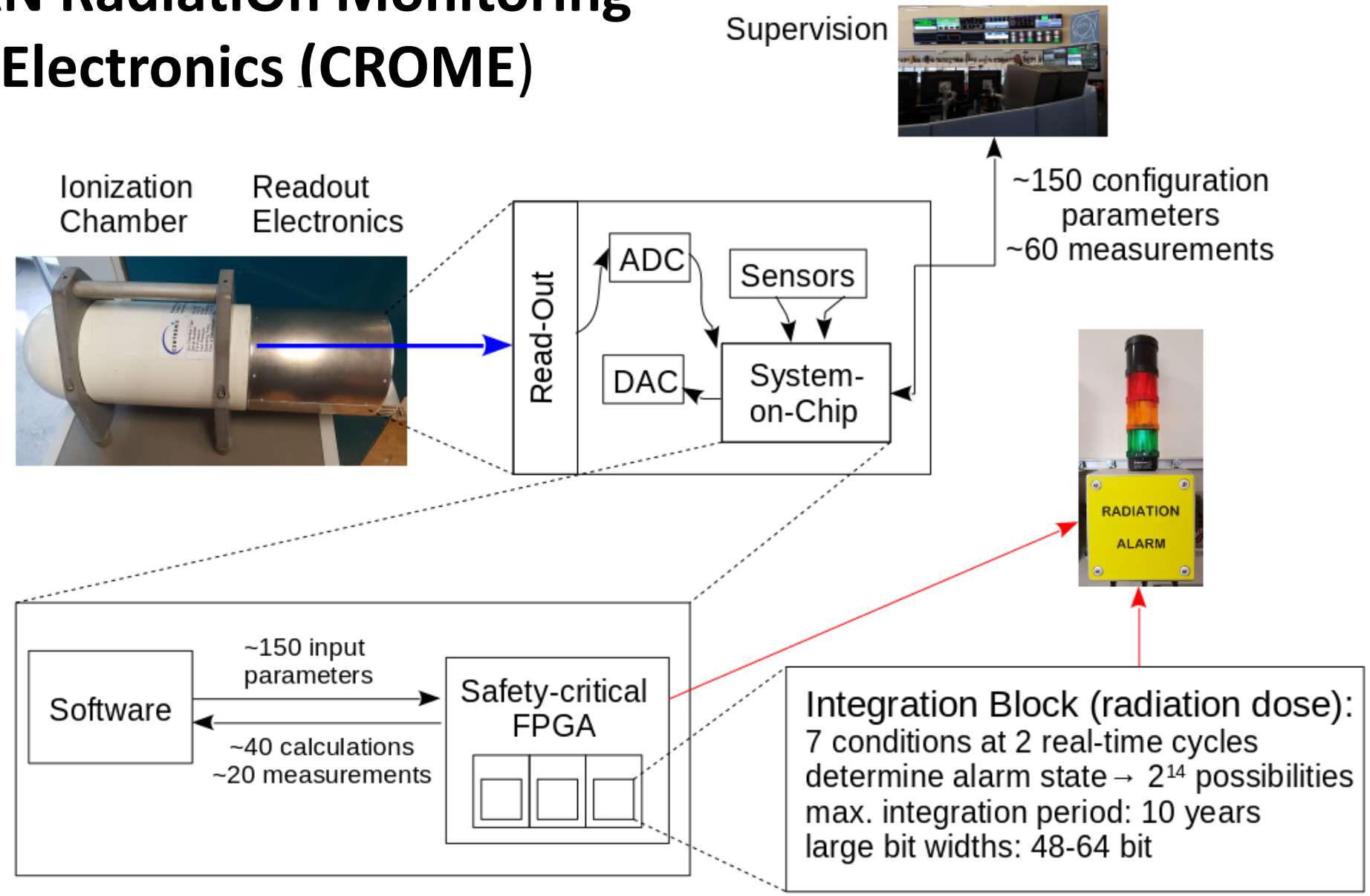
Life Cycle





Radiation Protection at CERN

CERN RadiatiOn Monitoring Electronics (CROME)



Natural Language Property Mapping

Natural Language	SystemVerilog
Expr1 equals Expr2	`Expr1 == Expr2`
Expr1 and Expr2	`Expr1 && Expr2`
Expr implies that: Seq	`Expr |-> Seq`
Every time when Expr: Seq	`Expr |-> Seq`
TimeSeq after Expr: Seq	`Expr ##TimeSeq Seq`
(Expr)	`(Expr)`
(“ ”)	

Application Specific NLP Mapping

Natural Language	SystemVerilog
Cycle is the start of a MC	`$rose(mtValidxDI)`
The time passed since integration start	`== etCount`
At the previous MC	`signalNameLastMC`
At integration end time	`elapsedTimexDO >= integralTimexDI`
Once the calculation is ready	`##`nrCUntilCalcRdy`
Alarm function is/was activated	`alarmActivexDI == 1`
Integral value	`signed'(integralxDO)`

MC = Measurement Cycle

Example 1

- <u>Requirement:</u>

  "A manual zeroing of the integrated value shall be possible."

- <u>SystemVerilog property:</u>

```
property pIntManualResetNextMT();
    ($rose(mtValidxDI) && intResLastMC == 1)
    |->
    ##`nrCUntilCalcRdy integralxDO == integralTB;
endproperty
```

MC = Measurement Cycle

Example 1

```
property pIntManualResetNextMT();
    ($rose(mtValidxDI) && intResLastMC == 1)
    |->
    ##`nrCUntilCalcRdy integralxDO == integralTB;
endproperty
```

- <u>Natural Language Property:</u>

  Cycle is the start of a MC and integral reset at previous MC equaled 1

  implies that:

  once the calculation is ready, integral value equals the testbench internal integral value (“integrated since last MC”)

MC = Measurement Cycle

Example 2

- <u>Requirement:</u>
- "It shall be possible to manually trigger a reset of an integration alarm through the supervision software."

- <u>Natural language property :</u>

```
"(Cycle is no MC and (alarm was configured as latched at the
previous MC) and alarm reset equals 1 and
(integral value is less than (threshold at previous MC) or alarm
function was deactivated at previous MC))
implies that:
(in one clock cycle, alarm is off)"
```

MC = Measurement Cycle

Example 2

"(Cycle is no MC and (alarm was configured as latched at the previous MC) and alarm reset equals 1 and (integral value is less than (threshold at previous MC) or alarm function was deactivated at previous MC))
implies that:(in one clock cycle, alarm is off)“

- <u>SystemVerilog property:</u>

```
property pIntAlarmResetBetweenMT1();
    (mtValidxDI == 0 && latchedLastMC == 1 &&
    integralAlarmResetxDI == 1 &&
    (signed’(integralxDO) < signed’(thresholdLastMc) ||
    alarmActiveLastMc == 0))
    |->
    ##1 (ALARMxDO == 0);
endproperty
```

MC = Measurement Cycle

Results for CERN RadiatiOn Monitoring Electronics (CROME)

- Found a severe safety-critical fault
  - Design and verification engineer had same assumptions
  - Requirements engineer intended different meaning
  - Review of NLPs discovered the discrepancy

- Review of NLPs for integration dose block took only 1 hour

Conclusion & Outlook

- Natural Language Properties (NLPs) & workflow

- Main goal:
  - Avoid misunderstandings
    - Reduce risk of faults
    - Reduce number of iterations during development

- Adds to V-Model:
  - Specification of NLPs
  - Review of NLPs by requirements engineer → validation

Conclusion & Outlook

- Found a safety-critical fault in the CERN RadiatiOn Monitoring Electronics (CROME)
- NLPs will be used for further verification of CROME
- Possibility for automated translation of NLPs into SystemVerilog

Thank You!

Questions?

accellera SYSTEMS INITIATIVE

2019 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION EUROPE