Infineon

Munich - November 18, 2013

Compatible Qualification Metrics for Formal Property Checking

Holger Busch
Senior Staff Engineer Verification
Infineon Technologies

Never stop thinking

Overview

- Motivation
- Goals
- Qualification Approaches
- Onespin‘s Coverage Feature
- Certitude
  - General set-up
  - Coupling with Onespin
- Experience and Comparison
- Conclusions

Why Qualify Formal Property Sets ?

- *"Formal properties are exhaustively checked! "*

➢ 100 % coverage ?

- Yes:
  - All input combinations implicitly checked by formal provers
- No:
  - Property assumptions constrain inputs
    (better than constrained randomized simulation: not just seed)
  - Property commitments cannot check all outputs
  - Single property cannot cover all input-/output behaviour
  - ➢ Properties are developed according to partitioned DUT-function

➢ Task: Guarantee completeness of partitioning

Goals

- Quality control
  - Assessment of formal property sets
- Formal verification management
  - Progress indication
  - Sign-Off Criteria
- Handling of mixed verification tool landscape
  - Directed & constraint driven randomized simulation
  - Formal property checking
- ISO26262 compliance of automotive µC products
  - Traceability
  - Documentation of design and verification process

Qualification Approaches for Formal

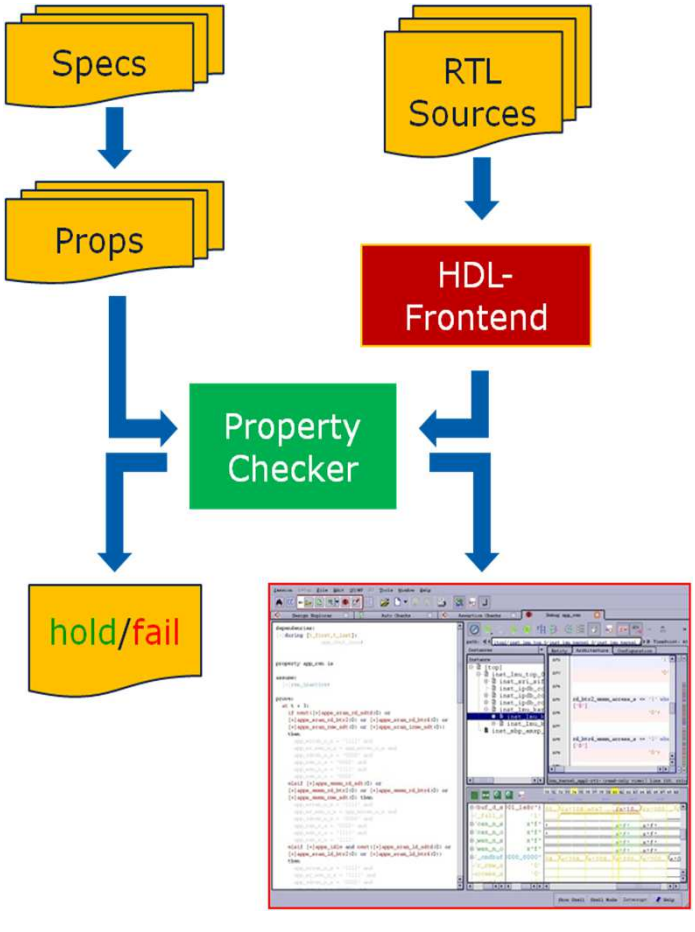
- Manual
  - Review of formal properties
- Formal completeness checks
  - Onespin‘s gap-free verification methodology
  - ➢ Strongest criterion, not related to simulation coverage metrics
- Formal witness generation
  - Simulation coverage for witness trace: line, branch
  - ➢ Quality of witness ?
- Design mutation
  - Onespin‘s built-in coverage feature Quantify
  - Link to test-bench qualification tool Certitude

Overview

- Motivation
- Goals
- Qualification Approaches
- Onespin‘s Coverage Feature
- Certitude
  - General set-up
  - Coupling with Onespin
- Experience and Comparison
- Conclusions

Onespin 360°™ MV

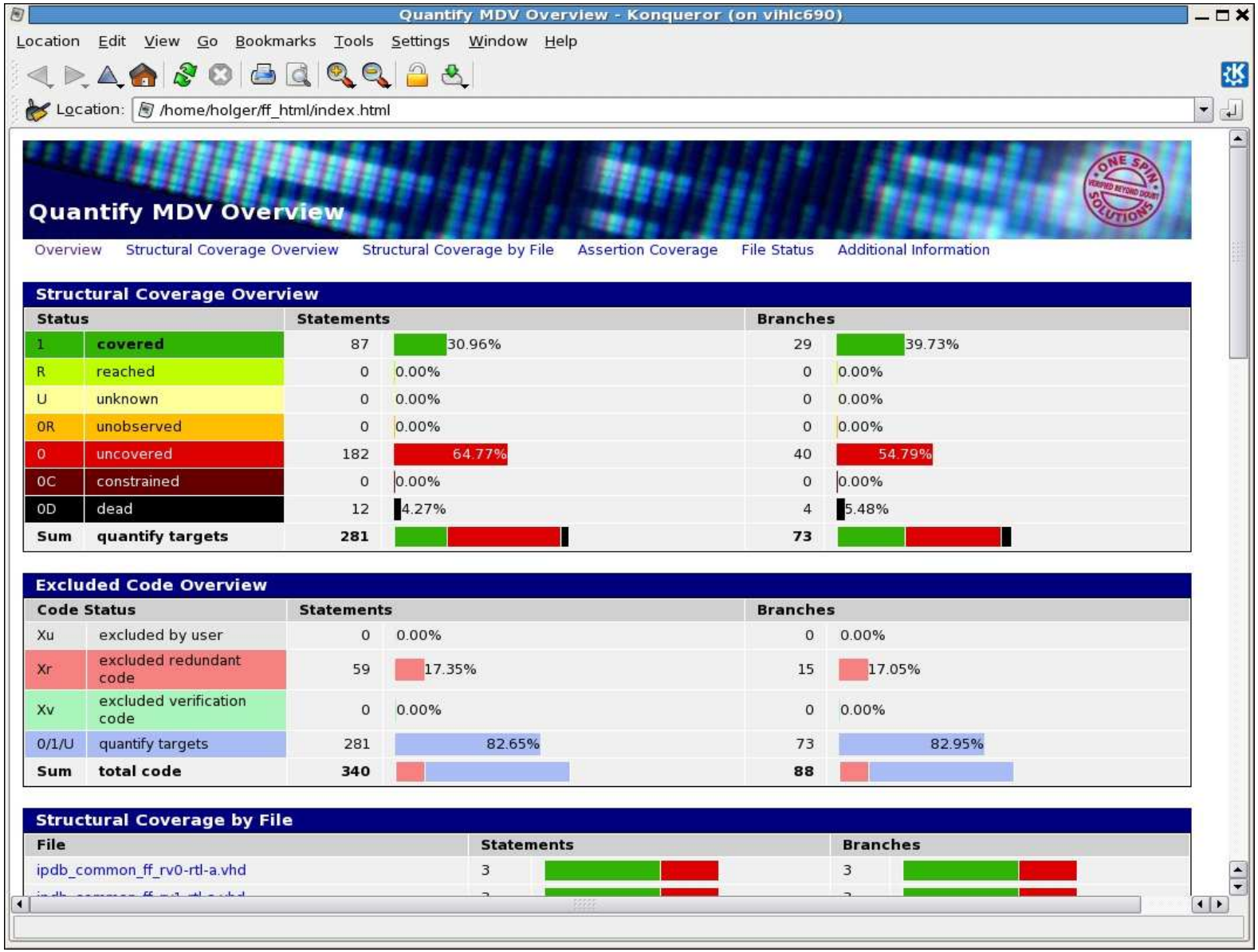
- Bounded model-checker
  - Various proof engines
- Property languages:
  - ITL (Interval Language) , SVA, PSL
- Linting
- Consistency checker
  - Dead-code, Stuck@signals, ...
- Property debugger
- Coverage
  - Formal completeness checker
  - Line & branch coverage: “Quantify“



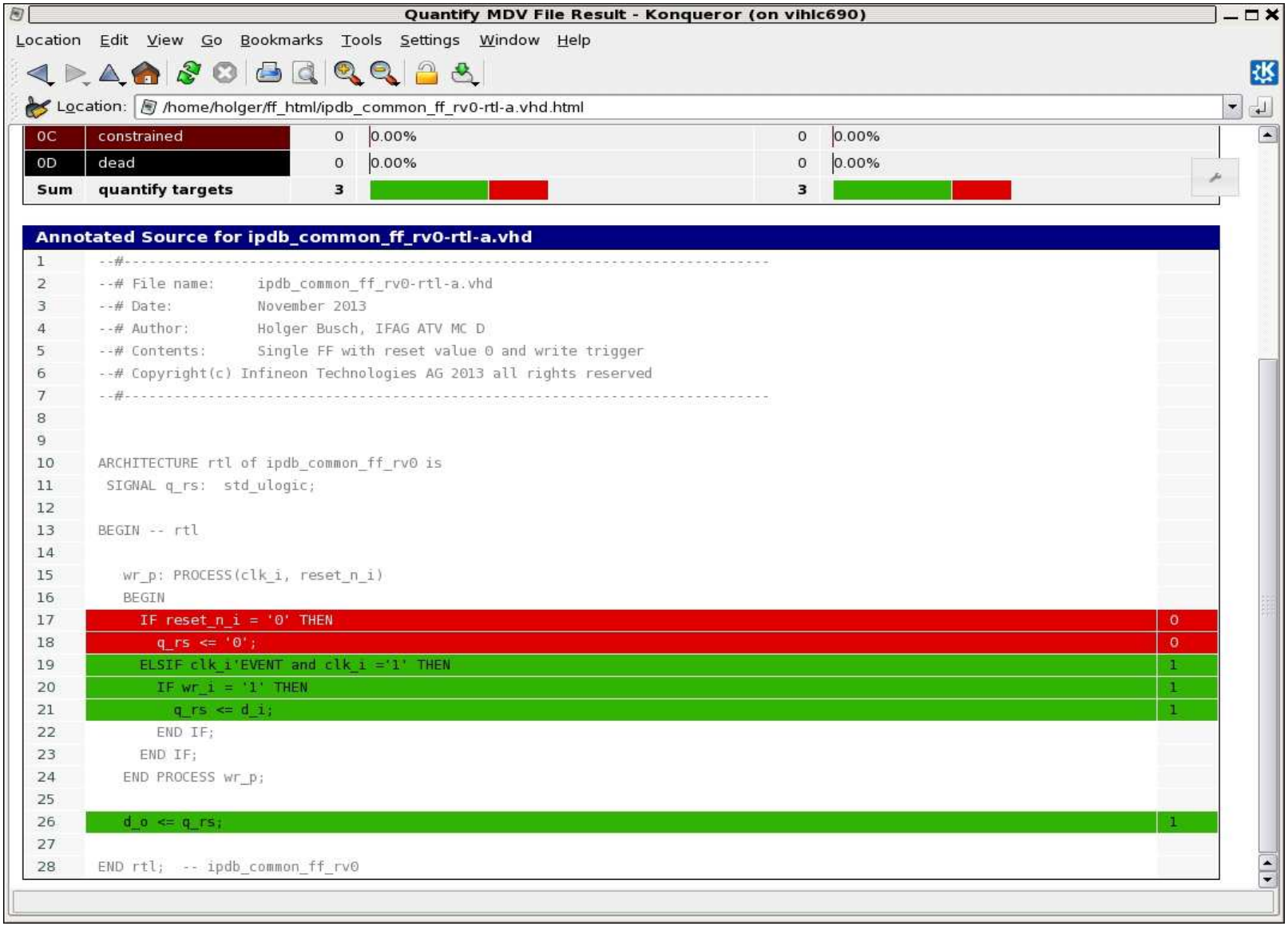
Onespin‘s Quantify Feature

- Pre-analyses
  - Formal-proof-based identification of dead, constrained, redundant code regions
  - Code reachability by given property set
- Observation coverage:
  - Formal proof that code location (assignment) checked
  - Code location observed when proof fails
- User-guidance easy
  - Push-button, focussing to code regions possible
- Result presentation
  - XML ~> UCDB-compatible
  - HTML-visualisation

Onespin's Quantify Feature



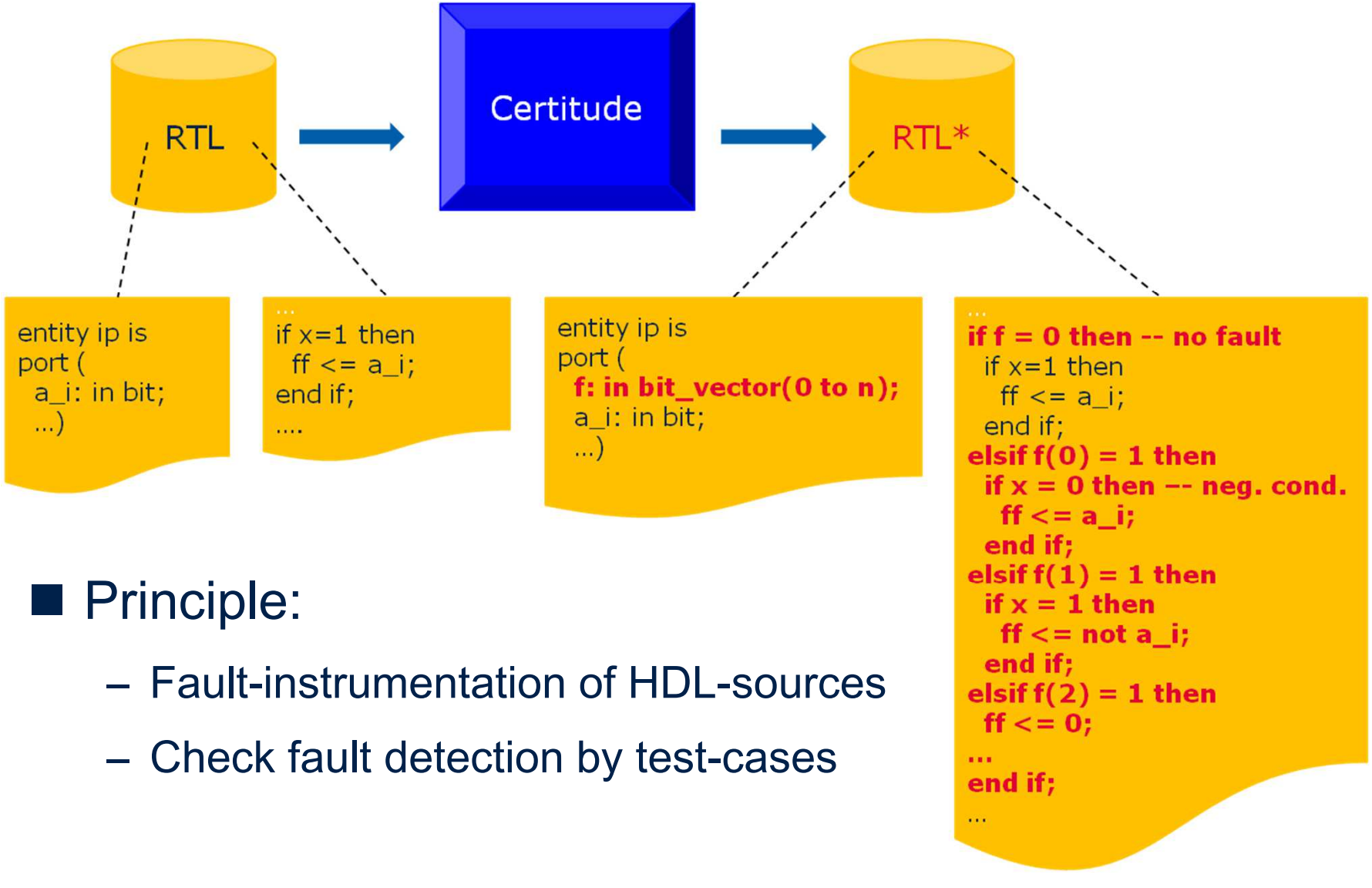
Onespin‘s Quantify Feature



Overview

- Motivation
- Goals
- Qualification Approaches
- Onespin‘s Coverage Feature
- Certitude
  - General set-up
  - Coupling with Onespin
- Experience and Comparison
- Conclusions

Certitude: Fault Instrumentation

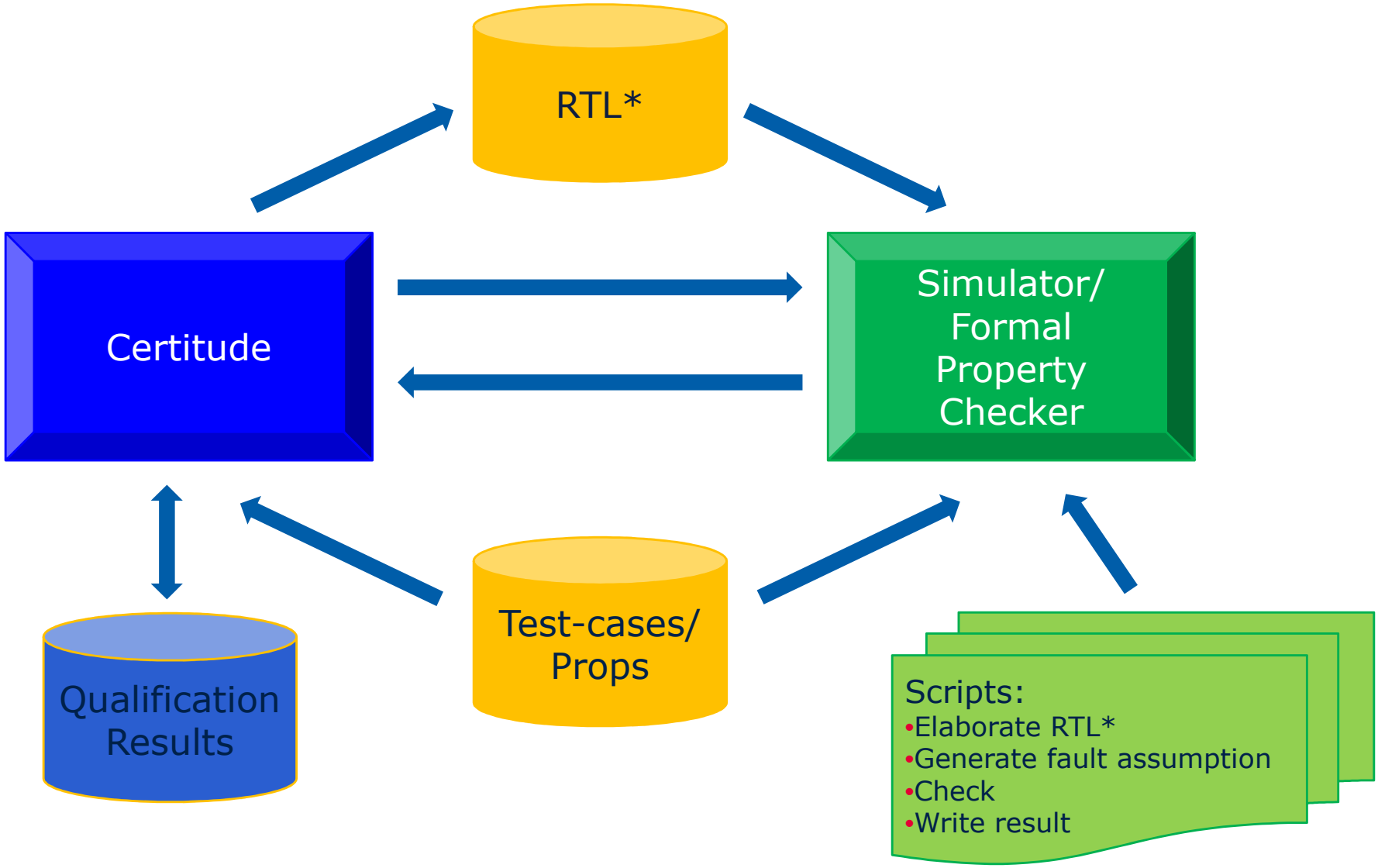


- Principle:
  - Fault-instrumentation of HDL-sources
  - Check fault detection by test-cases

Certitude: Qualification Phases

- Modelling phase: RTL-code instrumentation by Certitude
  - Different fault models injected into RTL code
  - Top-level entity with additional input vector for individual activation
- Activation phase: Each test-case run once:
  - Activation: test-case stimulus activates fault condition
  - Propagation: fault visible at observation points (DUT interface)
  - Detection phase: Analyses for pairs of {fault test-case}:
  - Detection: fail of test-case instead of pass
  - Fault-sets: $F_{injected} \supseteq F_{activated} \supseteq F_{propagated} \supseteq F_{detected}$
  - Iterative detection controlled by Certitude:
- Statistical Approach by Certitude:
  - Metrics computation for statistical samples
- Application to Formal Properties
  - Iterative invocation of property checker for {fault / formal-property} pairs instead of simulator for {fault / test-case} pairs

Certitude Qualification Flow



Overview

- Motivation
- Goals
- Qualification Approaches
- Onespin‘s Coverage Feature
- Certitude
  - General set-up
  - Coupling with Onespin
- Experience and Comparison
- Conclusions

Certitude <-> Onespin

- Iterative procedure:
  - Let Certitude select:
    - Property P from set of qualification properties
    - Fault c from current set of non-detected faults
  - Add fault assumption to regular property

Regular Property P:

$\forall c.\ f(c)= 0$
$ass(P) \vdash com(P)$

Property P with enabling of fault c:

$1hot(f) \wedge f(c)=1,$
$ass(P) \vdash com(P)$

  - Check fault-C-enabled Property P in property checker
  - Return proof result + run-time to Certitude
    - Fail: fault c detected by Property P
  - Repeat until Certitude is finished:
    - All faults detected

    or
    - All {fault / property}-pairs exercised for non-detected faults

Overview

- Motivation
- Goals
- Qualification Approaches
- Onespin‘s Coverage Feature
- Certitude
  - General set-up
  - Coupling with Onespin
- Experience and Comparison
- Conclusions

Experience and Comparison

Quantify	Certitude <-> Onespin Qualification
Added value: Coverage results useful for productive verification projects	
Design size: Modules with several 10 k locs manageable	
Usage: easy	**Usage:** less easy at the beginning: Set-up for Certitude and FPC required
Fault injection: elaborated model	**Fault injection:** RTL design
User control: Code regions (focus, skip, exclude) Property set	**User control**: rich configurabilty Code inclusion, fault types, density, properties; instance- or module-based
Restartability: yes (longer setup time)	**Restartability**: yes (short set-up time)
Compatibility with simulation: mergeable	**Compatibility with simulation:** 1:1
Maturity: Product feature, potentials for improvements	**Maturity**: Certitude available for many years; recently: scripting for IFX-internal usage for performance optimization
Licences: regular Onespin prover licenses	**Licences:** Onespin prover licenses + Certitude license

Overview

- Motivation
- Goals
- Qualification Approaches
- Onespin‘s Coverage Feature
- Certitude
  - General set-up
  - Coupling with Onespin
- Comparison
- Conclusions

Conclusions

- Two simulation-compatible FPC qualification methods for productive usage
- Both handle big IPs and property sets
- ➢ Usage strongly recommended!
- Two different paradigms:
  - Integrated in property checking environment: Quantify
    - Efficient, but closed and vendor-specific
    - Metrics similar to simulation coverage
  - Coupling of separate tools: Certitude-Onespin/Simulators…
    - Open for customization
    - Exactly same metrics for FPC and simulation
  - ➢ Potential synergies

Questions ?