A Structured Approach to verify Ties, Unconnected Signals and Parameters

Saurabh Singh, NXP

Peter Limmer, NXP

Thomas Luedeke, NXP

accellera SYSTEMS INITIATIVE

NXP

2016 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION EUROPE

Contents

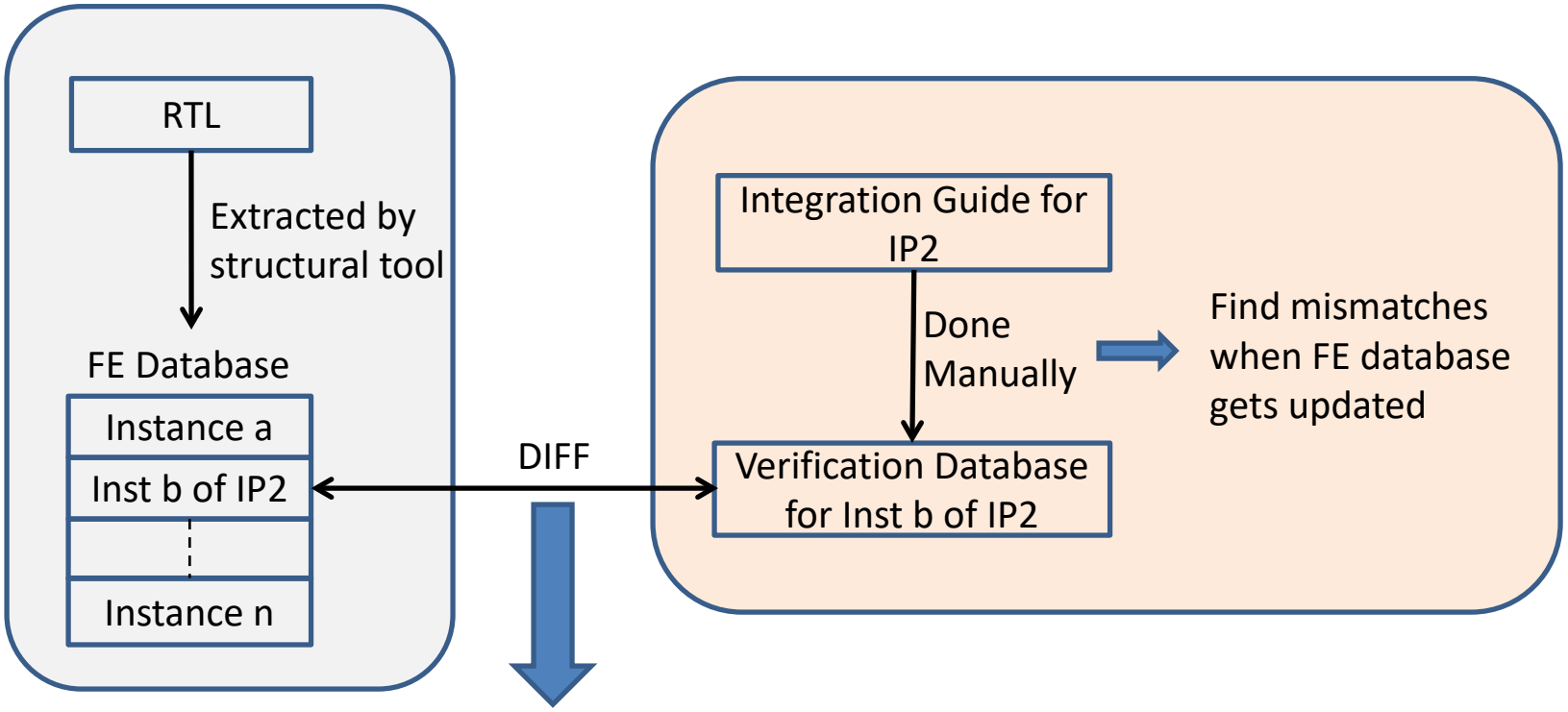
- Motivation
- Concept
- Automated checks: Input Format
- Automated checks: Output Format
- Interpretation of mismatches
- Frontend database generation flow
- Execution Flow
- Results & Conclusion
- Future Work

Motivation

To replace visual checks for unconnected signals/ ties/ parameters by automated checks

- Checks done automatically on every Frontend(FE) release
- Catch incorrect values of ties/parameters early in the design/verification phase
- Catch missing TUP (ties, unconnected signals, parameters) in integration guides
- Remove mismatches between RTL and integration guides
- Catch any change in RTL wrt TUP

Concept



- Missing TUP
- Incorrect values for ties/parameters

Automated Checks: Input Format

- Input in the form of text file!
- Three input files for each instance – one each for T,U,P
- TUP description format: signal name, index (if any), signal width, data type, value
- Data Type: pseudo Verilog style – hexadecimal/decimal/ binary
- No hierarchical path needed

Input Format Examples

- Tied signal “ipp_ind_pqspi”. Bits 4 to 11 are tied.

```
ipp_ind_pqspi[11:4] 8 d 0
ipp_ind_pqspi[11:4] 8 h 00
ipp_ind_pqspi[11:4] 8 b 0000_0000
```

- Unconnected signals

```
ipp_ana_en_sample[15]
ipp_decode_extch[2:0]
```

- Parameters

```
START_ADDR[15:0] 16 h 1700
ADDR_WD_PP[31:0] 32 d 14
CNT_BITS[31:0] 32 d 4
```

Automated Checks: Output Format

```
Instance name: instance_n

SIGNAL(S) MISSING IN FRONTEND DATABASE
Signal Name : signal_a
Signal Name : signal_b

SIGNAL(S) MISSING IN VERIFICATION DATABASE
Signal Name : signal_c
Signal Name : signal_d

VALUE MISMATCHES
Signal Name : signal_e
Value in frontend db : 3 d 1
Value in verif db    : 3 d 2

Signal Name : signal_f
Value in frontend db : 8 h 1
Value in verif db    : 8 h 5

SCRIPT_FAIL: 6 Error(s) detected !
```

Interpretation of mismatches

Type of mismatch	TUP present in RTL	TUP present in integration guide
TUP missing in FE database	No	Yes
TUP missing in verification database	Yes	No
Value mismatch	Yes	Yes

Mismatches indicate a bug in RTL or inaccuracy of integration guide.

FE Database Generation Flow

Step 1: Select the instance list as black box list

Step 2: Extract parameter info via structural checker tool

Step 3: Report black box outputs with no fanout via structural checker tool

Step 4: Report inputs to black boxes via structural checker tool

Step 5: Extract tied inputs from reported inputs

Execution flow for an instance

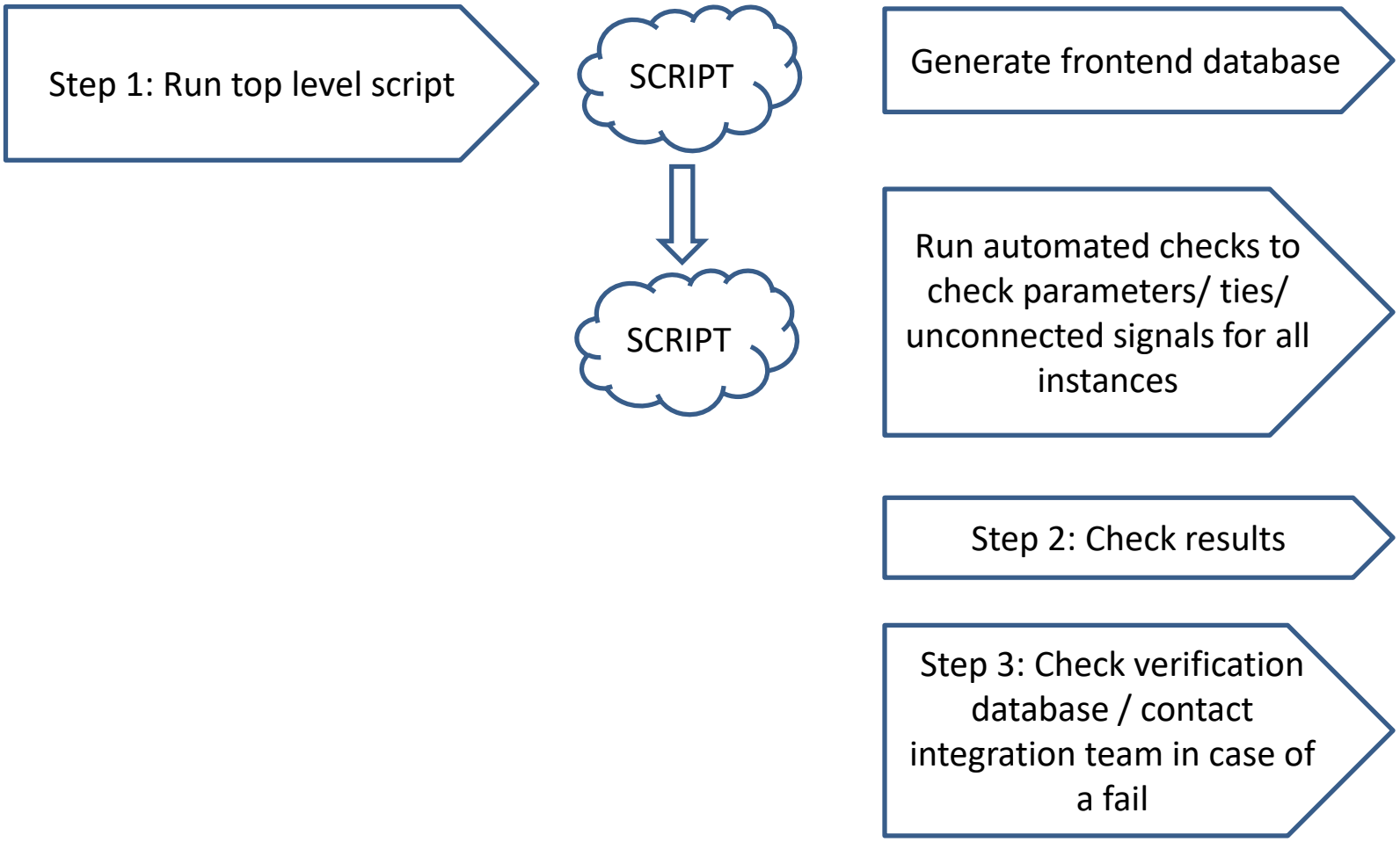
Step 1: Generate frontend database

Step 2: Run automated checks to check parameters/ ties/ unconnected signals for a particular instance

Step 3: Check results

Step 4: Check verification database / contact integration team in case of a fail

Execution flow for all instances



Results & Conclusion

- New **undocumented** TUP found in addition to:
  - Value mismatches
  - Incorrect TUP in RTL & documentation
- Formal tools don't find unspecified and unknown TUP
- Checks help to kick-off project
  - No need for testbench infrastructure for verifying TUP
- Exceptions for toggle coverage generated from same flow
- Automated checks compare RTL & documentation in both directions
  - Quality of documentation improved

Results & Conclusion

- No need to know verification tools like Cadence Incisive Formal Verifier (IFV)
- No need to know languages like C and SystemVerilog
- Checks can run independently or with regression
- Approach deployed successfully on two 32-bit SoCs
- FE Database generated once per given RTL database
  - Takes ~4 minutes per database
- Verification database created once per project
  - Takes few seconds to minutes per instance depending on complexity

Results & Conclusion

- Very fast turnaround time
  - Updating verification database and running automated checks takes few seconds
  - Debug time significantly less than formal or directed tests
  - No need to get simulation up and running

Future Work

- Setup to check default parameters
- Setup to detect inputs driven by constant value logic
- Improvement in reporting to track pass/fails

Questions