2018 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES

SoC Verification of Analog IP Integration through Automated, Formal-Based, Rule-driven Spec Generation

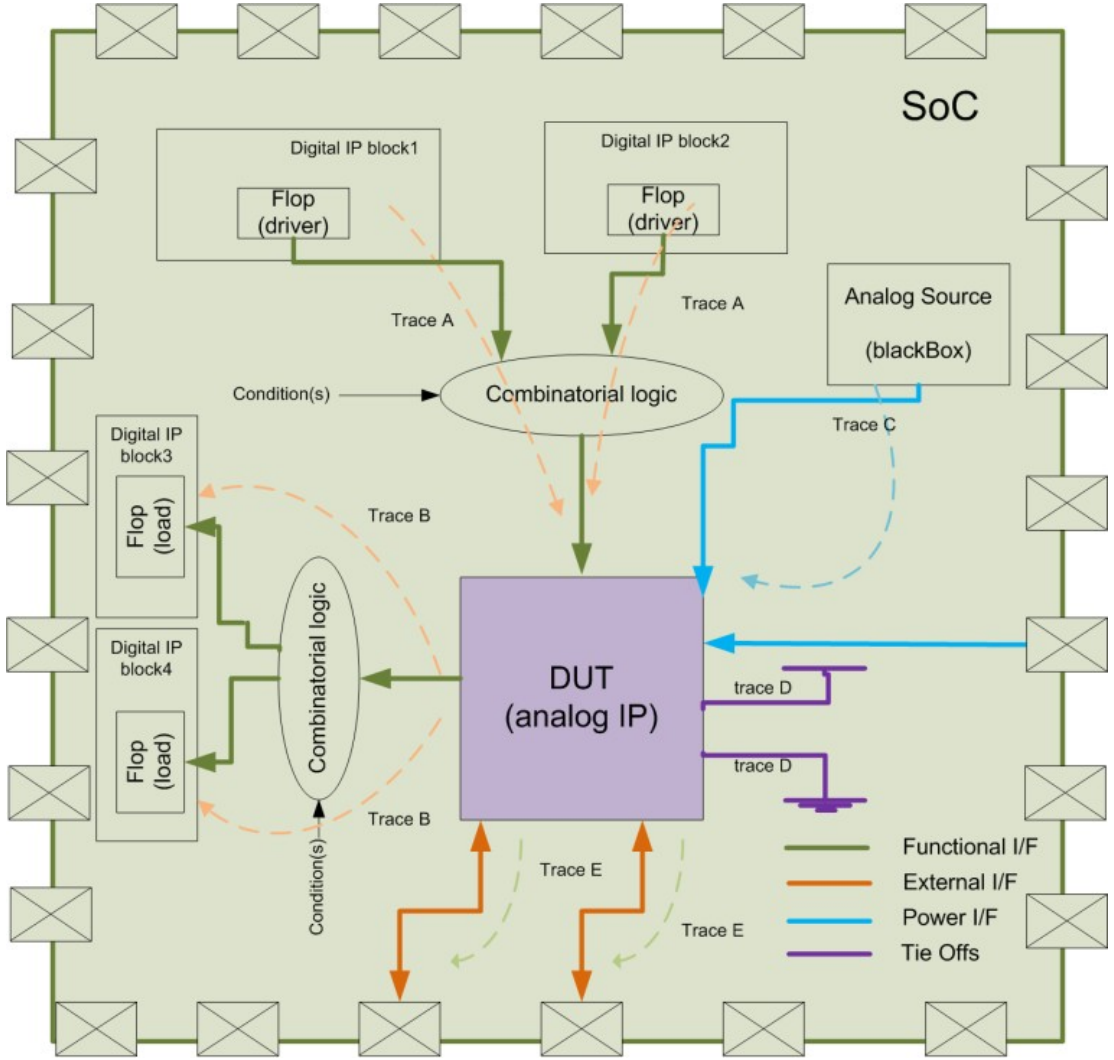
Murugesh Palaniswamy, Senior Staff Verification Engineer; Synaptics, Inc.
Ravi Kalyanaraman, Director of Design Verification; Synaptics, Inc.
Gargi Sharma, Application Engineer; Mentor, A Siemens Business
Bharat Baliga-Savel, FV Application Eng; Mentor, A Siemens Business

Our Connectivity Verification Challenge

- Numerous digital **and analog IP** blocks from internal and 3rd parties go into our SoCs
- Generation of the connectivity spec for each analog IP usually involves a lengthy manual process
- The problem is exacerbated by the restricted and not-so-thorough documentation from the respective analog IP providers

➢ Connection bugs can result in a re-spin!

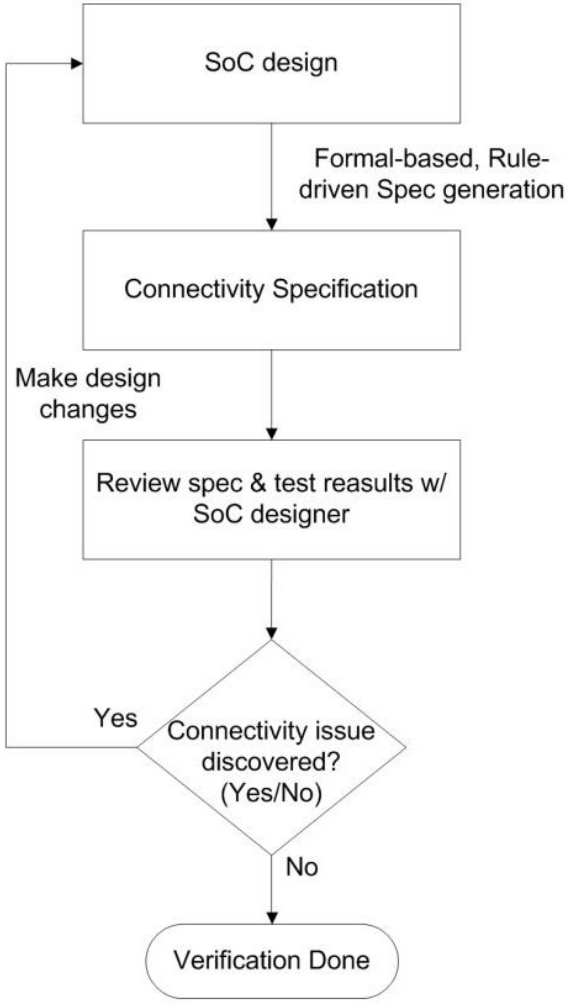
IP Connectivity Paths in a Typical SoC



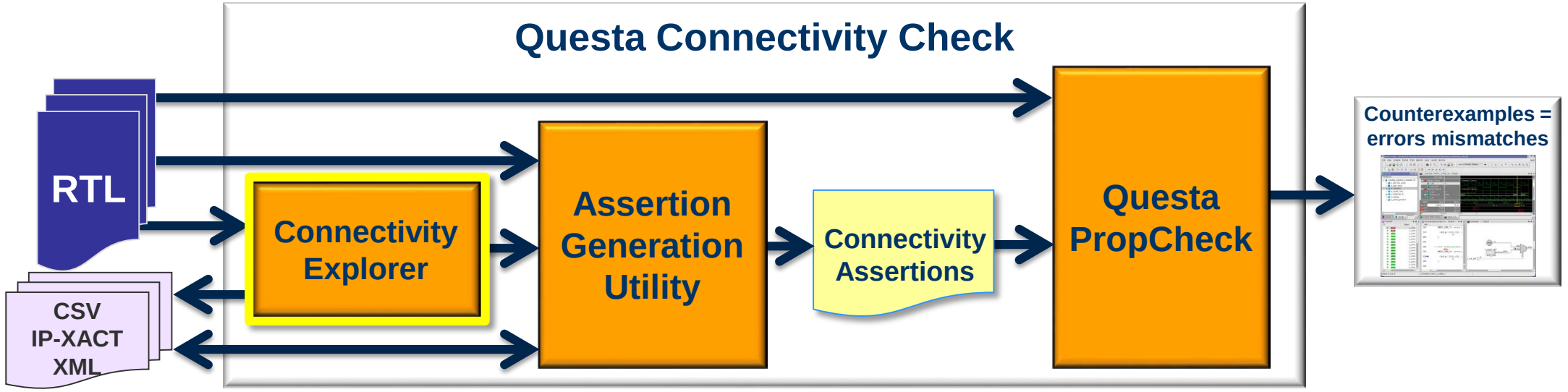
Limitations of the Prior Manual + Sim Flow

- Conventional approach: waveform tracing + RTL code tracing
  - Man-hours for this step increases linearly with the number of ports of the DUT
  - E.g. expect 12-14 verification man-hours for each IP with an average size of 100 ports
- Creating the testbench and tests was time consuming too
  - We able to automate much of this, but it required constant maintenance
  - Scripting minimized effort and time required so test generation became independent of size or number of ports on the IP
- Running the tests takes considerable compute resources & results analysis
  - Scripting can help flag obvious failures
  - Overall, the simulation and review cycle would take weeks of tedious, error prone work.

Our New Formal-Based Connectivity Specification Generation Flow



1 of 4: Formal-based, Rule-Driven Spec Generation



- Start by accepting that the baseline, fully assembled SoC probably has as-yet undiscovered issues!
- The current RTL snapshot is the baseline “spec”
- Connectivity Explorer automatically derives static & dynamic connectivity from the RTL

More on Formal-based, Rule-Driven Spec Generation

```
:
:
-inst<instance_name>: traces the connectivity of this particular instance of the analog IP
-sig <name>: traces both sides of the connectivity of signal/port belonging to the above
instance
:
:
```

- **Input spec format** lists the hierarchical start points and the direction of traversal (i.e., from core to analog IP or from analog IP to core)

- **Path traversal rules** specify rules that control the traversal
  - Navigating through junction points in combinatorial logic
  - Navigating through flops,
  - The stopping condition for the traversal etc.

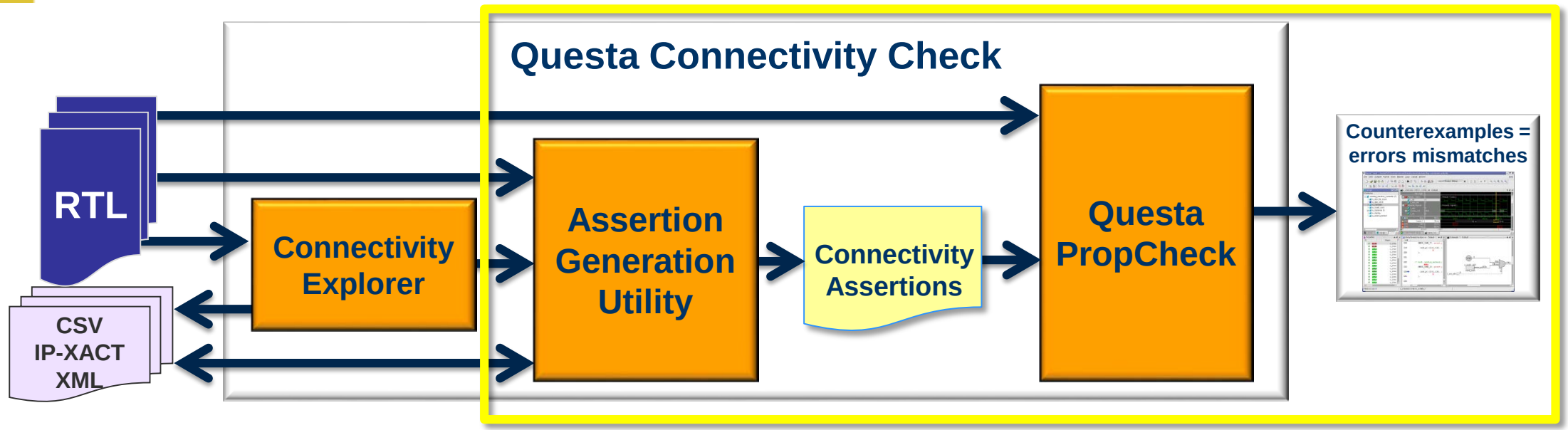
2 of 4: Creating & Reviewing the Connectivity Specification

type	src	dest	cond	delay
#Registers/Latches				
connect_dly	SOC_top.clk_top_U0.pll_wrapper_U0.frac_pll_U0.FMOD_pin	SOC_top.reg_blk_U0.regsOut[786]		1
#BlackBox Ports				
connect_dly	SOC_top.clk_top_U0.pll_wrapper_U0.frac_pll_U0.TP	SOC_top.pcie_phy_U0.TP		
#Primary Ports				
connect	SOC_top.ATP	SOC_top.clk_top_U0.pll_wrapper_U0.frac_pll_U0.TP		
#Constants				
tied_high	SOC_top.clk_top_U0.pll_wrapper_U0.frac_pll_U0.AVDD			
tied_low	SOC_top.clk_top_U0.pll_wrapper_U0.frac_pll_U0.AVSS			
#Instance Ports				
connect	SOC_top.analog_blk_U0.REFCLK	SOC_top.clk_top_U0.pll_wrapper_U0.frac_pll_U0.ref_clk_pin		

- After the first pass of the automated spec generation we perform a brief manual review to spot blatant issues that can be quickly corrected

- Rerun the spec extraction and generation utility to produce a “golden” baseline spec that we feed into the formal analysis part of the flow

3 of 4: Running the Formal Analysis



- Going forward use the regular Connect Check formal app flow to detect errors as new IPs are added
- Any mismatches between the spec and DUT are clearly defined by counterexample waveform(s)

4 of 4: Results Processing and Review

- When the connections match the spec – if the connectivity properties are formally proven – results will be reported in a log file

- Detection of errors by this flow is illustrated in waveforms called "counter examples"
  - If port A is supposed to be connected to port B, but it's connected to port C, the tool will generate a waveform that shows how the signal emanating from port A can activate port C.
  - **The tool delivers a clear, "root cause" result, which makes debug and fixing of issues very straightforward!**

Real World Results

- The new process flow was validated on an SoC having approximately 25 analog IPs, and over 200MG in scale
- Reduced the connectivity verification time from 120-160 hours down to 3-4 hours -- **a 95% savings!**
- Much higher quality of results due to the exhaustive nature of the formal analysis performed under-the-hood
- Very easy to set up for new projects, and does not require knowledge of formal or assertions

2018 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES

Summary

- The demands of mixed analog and digital IP connectivity spec and implementation verification now exceeds what a testbench simulation flow can effectively support
- Our new formal-based flow is easy to setup, runs 95% faster then the old flow, and provides exhaustive analysis
- The formal-based flow is now our plan of record!