FUJITSU

shaping tomorrow with you

Achieving Sign-off Efficiently Via Static Verification Tools Customized for High-Value Failure Modes

Atsunori Machida, Akitsugu Nakayama, Masashi Shibata
Katsumi Imamura, Yuji Yoshitani

Fujitsu Kyushu Network Technologies Ltd.
June 3rd 2019

Our Company Profile

Name	**Fujitsu Kyushu Network Technologies Limited (QNET)**
Executive	**President & Representative Director** **Katsuhiko Tanahashi**

[Established] 1st July 2005 (QDT: 1985)
[Offices] Headquarter
Fujitsu Kyushu R&D Center, 2-1, Momochihama 2-chome,
Sawara-ku, Fukuoka, 814-8588, Japan
[Stockholder] 100 million yen(wholly-owned by Fujitsu., Ltd.)
[Employees] 865 (as of Apr 2018)
[Certifications]
Quality Management System ISO9001:2015 (Certified)
Environmental Management System ISO14001 (Certified)

Our Company Profile (Business)

Operations Management Services

Network System Solution

Operational Administrative

Service Infrastructure

IoT/M2M Services

IoT：Internet Of Things
M2M：Machine to Machine

Optical Device

Optical/ Packet Network System

Transmission Switching

Development Technology
・ASIC/FPGA Design・Verification
・System Design・construction
・Hardware/Firmware/Software

Wireless/ Mobile Network System

Sensing

WIRELESS

Advanced Core Technologies

NETWORK | SIGNAL | IMAGE VIDEO | VOICE | AI

Application Services

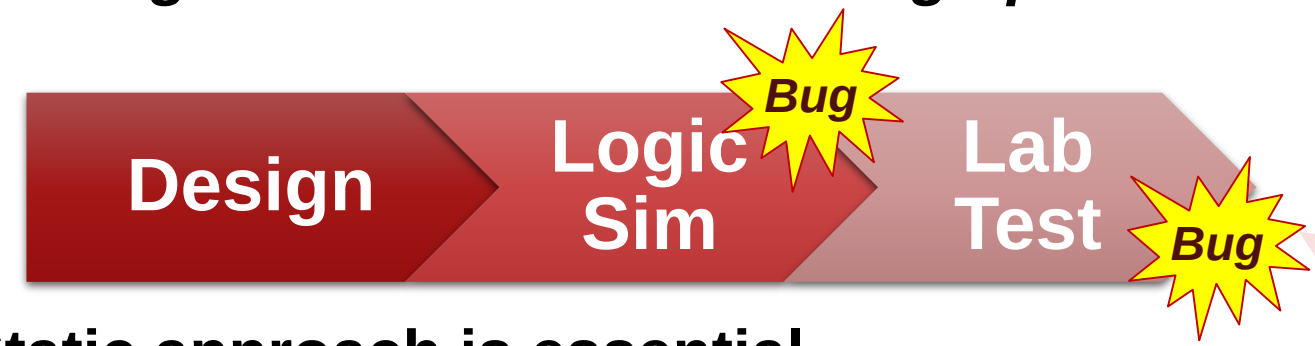
Research and Development of Hardware, Firmware and Software from Devices to Network Services

Background ~*Static Approach is Required for Efficiency*~

SOC logic scale has become large and complex

- **100s of IPs in SOCs**
- **100s of Clock Domains**
- **Huge amount of verification is needed**

➢ ***Bugs are missed in the design process***



Static approach is essential in early debug and for quality improvement

To find the following failure types early:

- ➢ **FSM, coverage, invariants**
- ➢ **FIFOs & Interfaces, Data-Control, Reset synchronization, Reconvergences, Clock-gating**
- ➢ **Glitches, Gray Code violation, Data-Stability violation, Pulse-Width violation, and so on**

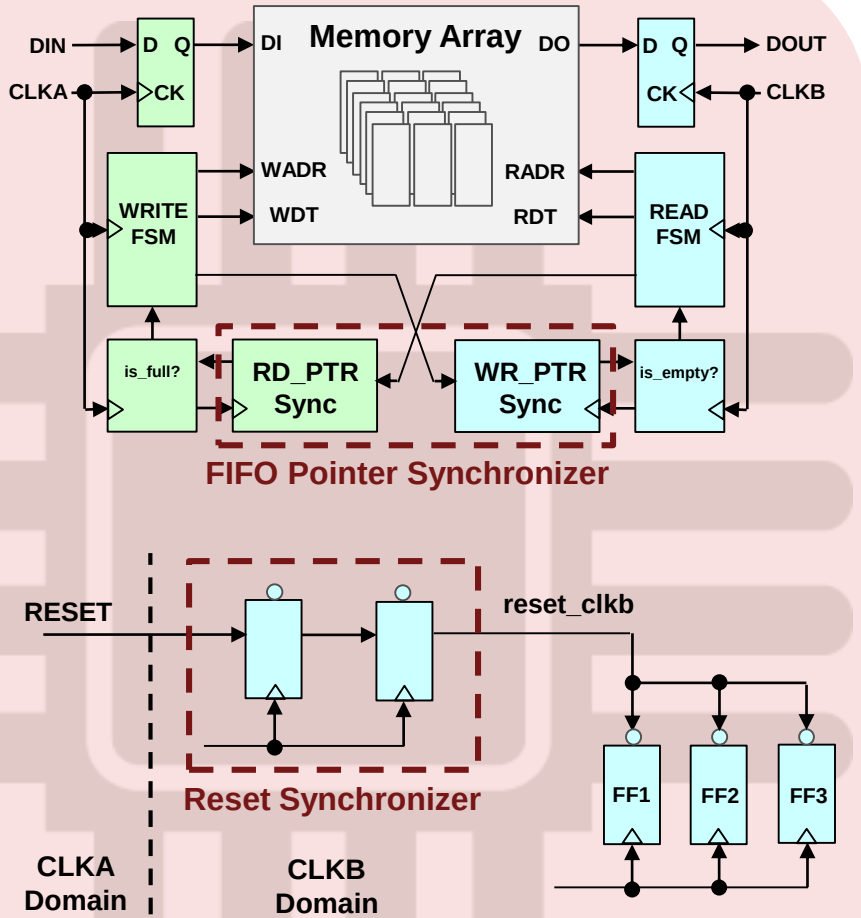


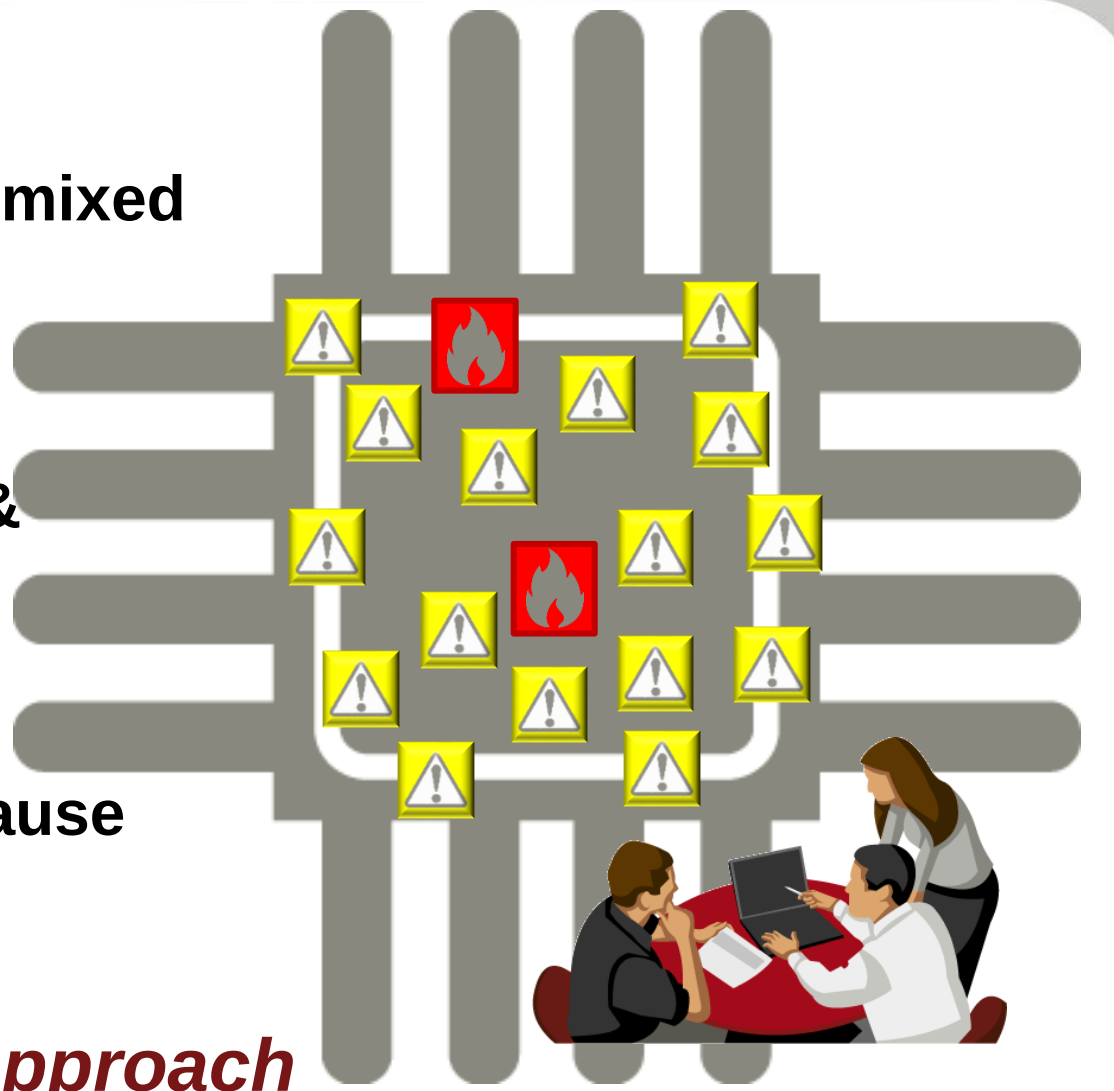
Figure 1: CDC Design Idioms

Motivation *~For Effective Static Approach~*

- **Early RTL Sign-off:**
  - **100,000+ checks!**
  - **Failure symptoms and root-cause gets mixed**

- **CDC Sign-off:**
  - **100s of Clock Domains!**
  - **CDC tool requires specific knowledge & skills for precise analysis**

- **Both Early RTL Sign-off and CDC Sign-off:**
  - **Designers fix the symptoms not root-cause**
    => *Excessive Design Iterations*



Upshot: Bugs missed even in static approach
=> Require Effective Static Approach to solve this problem

Main Idea *~Tool and Method for Efficient Static Method~*

Goal: Introduce Effective Tool and Establish Systematic Method

- **Systematic Method Requires**
  - **Good Tool-Setting Guidance**
  - **Effective Failure-Prioritization**
- **Real Intent's Verification Tool has**
  - **Setting Check Function**
  - **Prioritized Failure Listing Function**
- **Evaluated tool usability and efficiency in a trial project case study**

REAL INTENT

ASCENT

AutoFormal

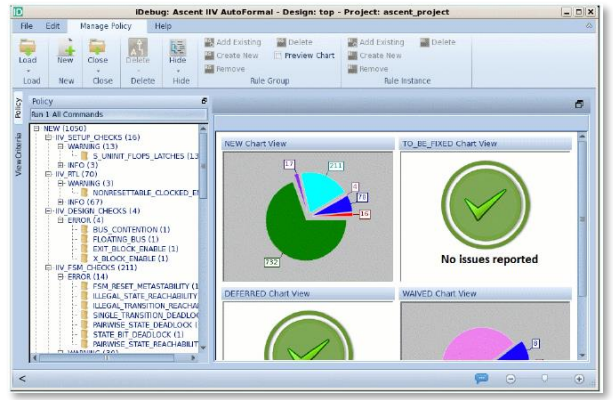


Figure 2: ASCENT GUI Example ©Real Intent, Inc.

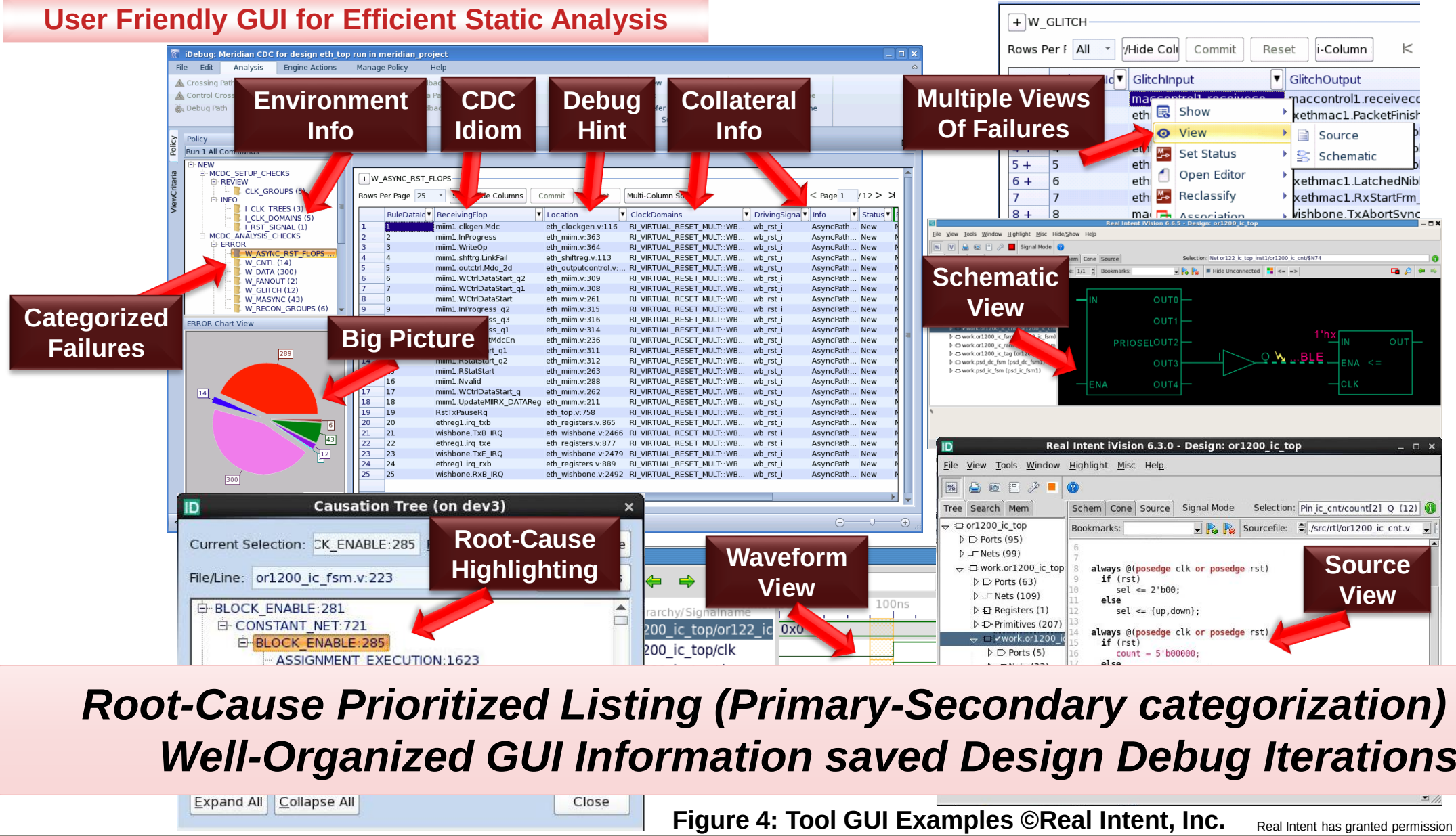
MERIDIAN

CDC

Figure 3: MERIDIAN GUI Example ©Real Intent, Inc.

Impression ~*Tool Organizes Information Smartly for Easy Debug*~

User Friendly GUI for Efficient Static Analysis



Root-Cause Prioritized Listing (Primary-Secondary categorization) and Well-Organized GUI Information saved Design Debug Iterations

Figure 4: Tool GUI Examples ©Real Intent, Inc.

Achievement ~*Efficient Early RTL Sign-off via Effective Static Tool*~

Efficient Early RTL Sign-off

- **Whole-chip analysis is achieved**
- **Found critical deadlock in FSMs in 2 projects**
  - In 3rd-party RTL => revelation to Fujitsu designers!
- **A 30% reduction in logic simulation TAT**
  - Primary-Secondary listing saved design iterations
  - Huge compression of items to review
- **Performed focused checks on RTL patterns**
  - Behavioral control
  - FSMs
  - Tristate drivers

Tool Execution Example on One Project

Whole Chip Analysis is Completed!

Some other tool gave up (Exploded)

Tool engine is customized for specific failure mode such as Auto Formal verification and CDC. Analysis went well.

List1: Simple Example of Failure Result for 106,356 Logic-gate SOC

	ERROR (Primary)	WARNING (Secondary)	INFO
DESIGN CHECKS	6	0	98
FSM CHECKS	1	9	348
LANGUAGE	0	0	31
COVERAGE	397	7214	92674

Designer could solve FSM issue by solve only one error debug Other tool detected these as 10 errors (Not 1 error and 9 warnings)

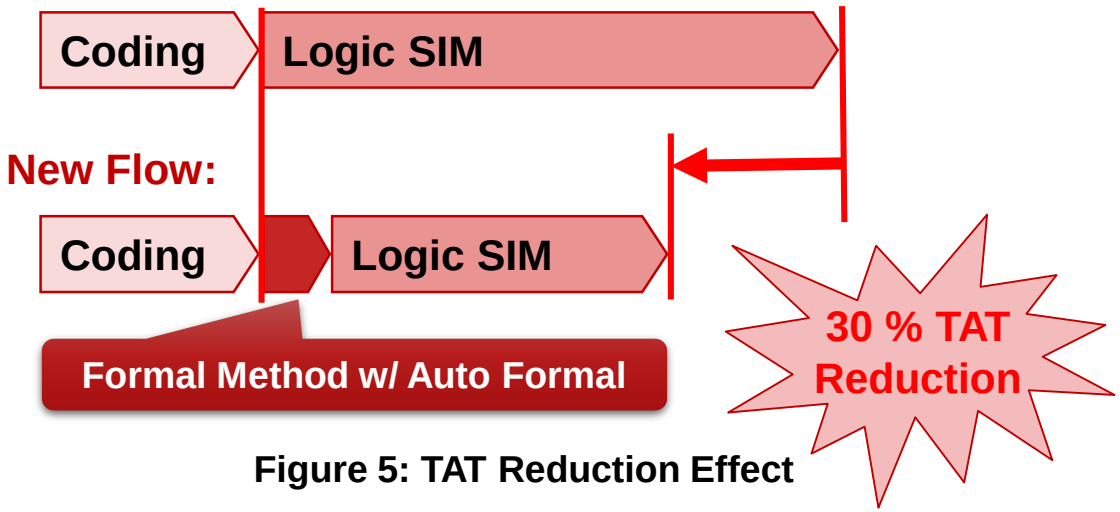


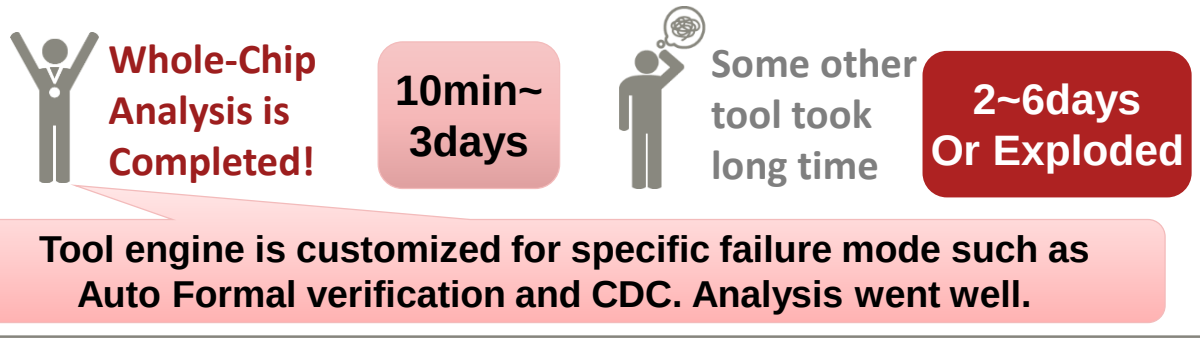
Figure 5: TAT Reduction Effect

Achievement ~*Systematic and Precise CDC Sign-off via Static Tool*~

Systematic and Precise CDC Sign-off

- **Whole-chip analysis is completed very fast!**
- **Setup check helped CDC analysis setting**
- **Recon is classified precisely**
  - Saved *weeks* of debug effort
  - Helped identify acceptable-recon vs error-recon
  - Real issues identified, and not missed because of volume of reporting
- **Failure is categorized against debug action type**
  - Debug actions for data-control association, reset-synchronization, recon are very different
  - Actions are very intuitive by violation categorization

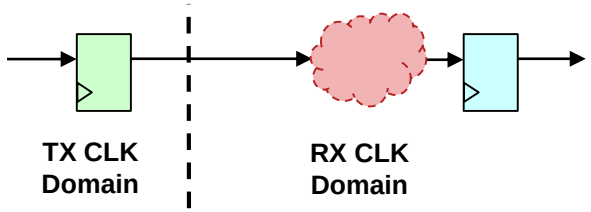
Tool Execution Example on One Project



Tool engine is customized for specific failure mode such as Auto Formal verification and CDC. Analysis went well.

Tool Ran Fast and CDC Failure was detected Precisely with Smart Failure Listing

- **Identified Failure Example:**
  - Missing Synchronizer



  - Missing Recon Interface

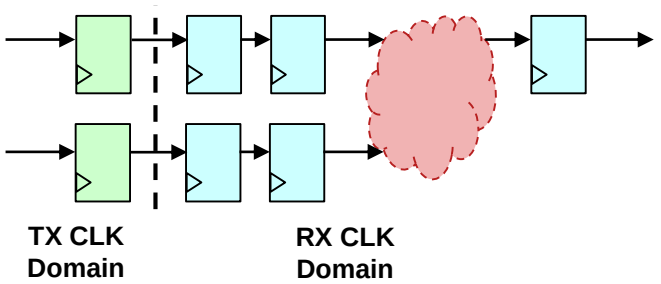
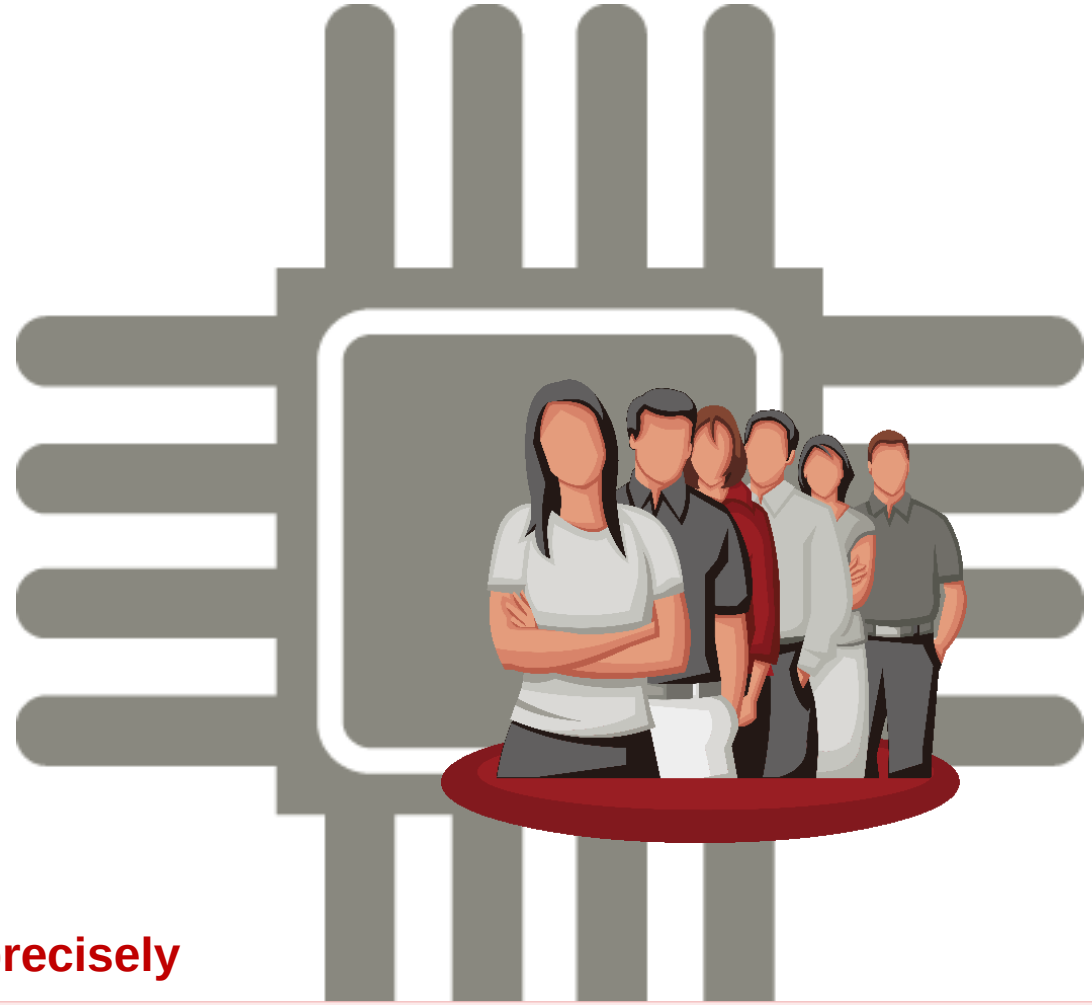


Figure 6: CDC Failure Example

Result ~*Tool Introduction itself became Efficient Static Methodology*~

Established Systematic Static Method just by Introducing Effective Static Tool

- Setup check helped correct tool setup
- Well-organized GUI and Root-cause prioritized failure listing saved design debug iterations
- Very fast static analysis was achieved by tool engine customized for each failure mode such as Auto Formal verification and CDC check



Systematic static method gave us large productivity and quality improvement

We could dispose bugs in an early phase quickly and precisely

To make the new systematic method better, we will continue to apply this approach and provide our experience-based feedback to tool vendor

Conclusions

- **Static approach is essential for robust sign-off**
  - **Early RTL sign-off and CDC sign-off are iconic examples**
  - **Complex High-end Computer and Networking SOCs require systematic static sign-off**
- **Effective static tool introduction itself became systematic sign-off methodology**
  - **Tool helped setup checking and root-cause analysis prioritized our debug approach**
  - **Very fast static analysis was achieved by tool engine customized for each failure mode**
  - **We have tested a systematic static approach that has enabled very high confidence in Early RTL sign-off and CDC sign-off**
- **Said approaches can become state-of-the-art default flows for all high-end SOCs**
- **To make the new systematic method better, we will continue to apply this approach and provide our experience-based feedback to tool vendor**