Accelerating and Improving FPGA Design Reviews Using Analysis Tools

Abdelouahab Ayari, Mentor, a Siemens Business, Germany
Anna Tseng, Mentor, a Siemens Business, USA
Kurt Takara, Mentor, a Siemens Business, USA

accellera SYSTEMS INITIATIVE

Mentor® A Siemens Business

2020 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION EUROPE

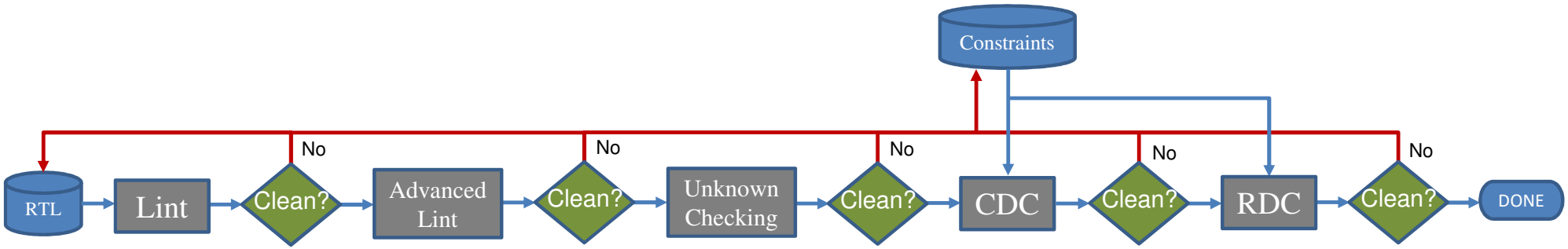
FPGA Design Quality and Reliability

- FPGA systems requirements
  - Functional safety
  - Reliability
  - Data integrity
- Challenges
  - Manual RTL reviews inefficient and not scalable
  - Target known risks with review checklists
  - Reviews time consuming and error prone
  - Reviews under schedule and budget pressures

Automated FPGA Review Flow

- Built upon design analysis tools
- Checks for latest industry standards and best-practices
- Automation provides high performance, low error, consistency
- Enables a well-defined, repeatable review process

Automated FPGA Review Flow



- Improves design quality
- Improves design review efficiency
- Enables faster and more consistent design review completion

Lint Analysis

- Automatic checking of RTL code for errors
- Checks against industry best practices
- Identify design elements problematic for FPGA mapping
- Identify problematic constructs
  - Ambiguous code
  - Incomplete sensitivity lists
  - Combinational loops
  - Incomplete state machines
  - Underflow and overflow conditions
- Enforce compliance to industry or project standards
- Early examination of RTL at syntax, semantic and structural levels

Lint Example

- Manual code reviews are tedious and error-prone
- The designer wants A=4
  - Needs a check for missing parens

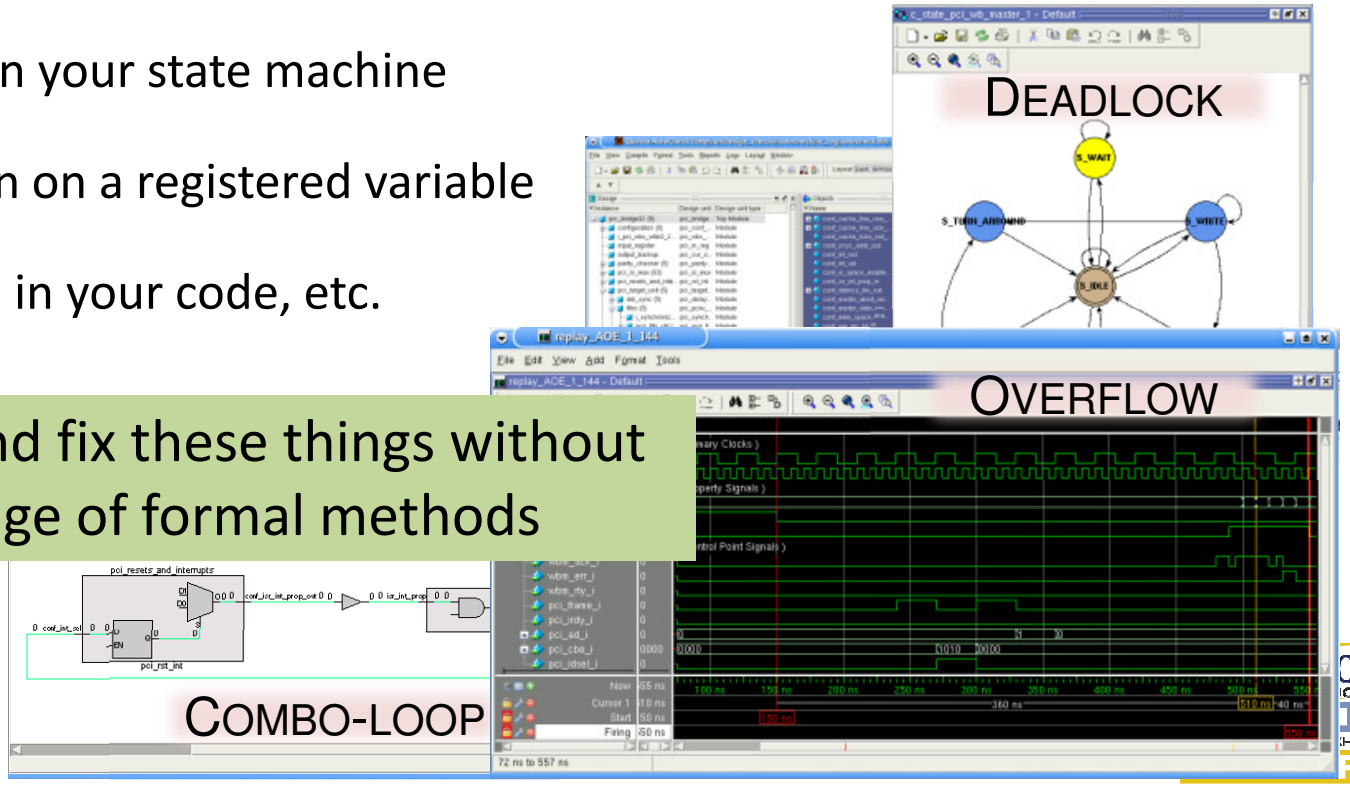
```
wire [3:0] a, b, c, d;

assign b = 4'h2;
assign c = 4'h1;

assign a = 4'h8 >> b >> c;
```

Advanced Linting

- Deep sequential checks using advanced formal technology
  - A deadlock scenario in your state machine
  - An overflow condition on a registered variable
  - A combinational loop in your code, etc.



The designer can find and fix these things without a testbench, or knowledge of formal methods

accellera SYSTEMS INITIATIVE

Advanced Lint Example

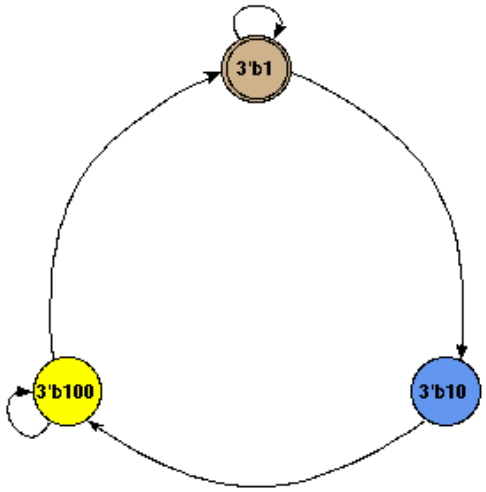
- FSM deadlock from incorrect structure
- Sequential logic causing FSM deadlock and dead code

Complex Sequential Logic

sig

Stuck sig => FSM deadlock

```
case (cstate)
  3'b001: if (en)
        nstate <= 3'b010;
    else
        nstate <= 3'b001;
  3'b010:
        nstate <= 3'b100;
  3'b100: if (sig)
         nstate <= 3'b001;
      else
         nstate <= 3'b100;
  default: nstate <= 3'b001;
endcase
```



Checking for Unknowns

- If unexpected ‘X’s appear on a critical signal or in an important register, your design can malfunction
- The circuit startup sequence or low power entry/exit can create cases where ‘X’s could corrupt critical design elements or signals
- The difference in the handling of ‘X’ semantics in synthesis and simulation can mask ‘X’ propagation issues

Challenges with X-States

Simulation to Silicon Mismatches

X-pessimism

- Simulation propagates X-state
- Should be a known good logic value

- Example: re-convergent fanout

```
regA <=  (!sel & i0) || (sel & i1);
```

X-optimism

- Simulation will propagate a deterministic value
- Should be X-state

- Example: conditional logic

```
if (sel)
    regA <= 0;
else
    regA <= 1;
```

X-State Analysis

Missed X-state bugs will result in bad silicon

'X' Risk Factors

- ❑ Circuit start-up & initialization
- ❑ Low power-related optimizations
- ❑ Multi-mode operations
- ❑ Gate level register state predictability
- ❑ Simulation vs. synthesis semantics

X Corruption

- ✓ **From all X src**
- ✓ **Check registers**
- ✓ **Check control**

Unknown Verification Benefits

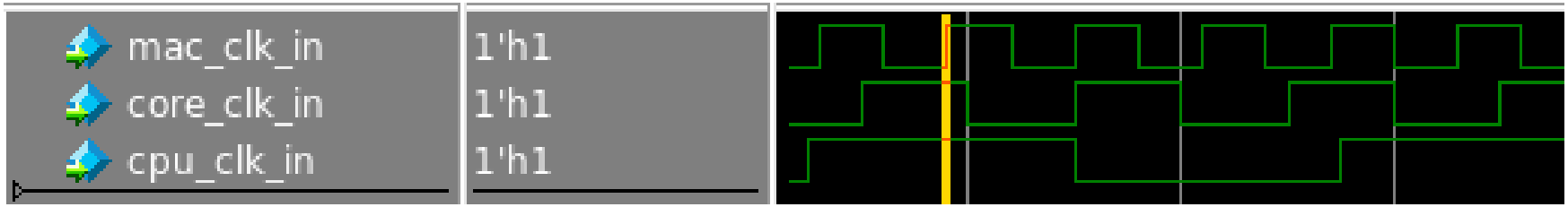
- ✓ Exhaustively identify all X-state issues
- ✓ Enables exhaustive evaluation of all circuit start-up and post-reset 'X' issues
- ✓ Fully automated analysis flow utilizing advanced formal technologies

accellera SYSTEMS INITIATIVE

2020 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION EUROPE

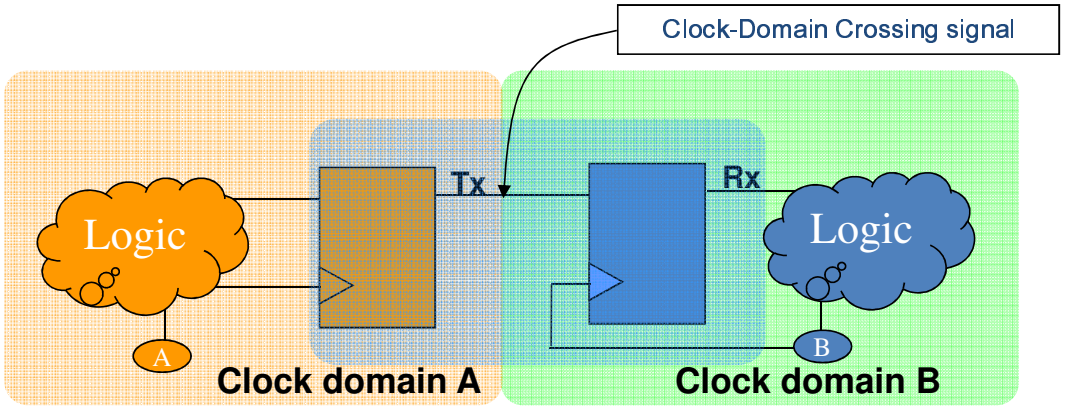
Clock-Domain Crossing (CDC)

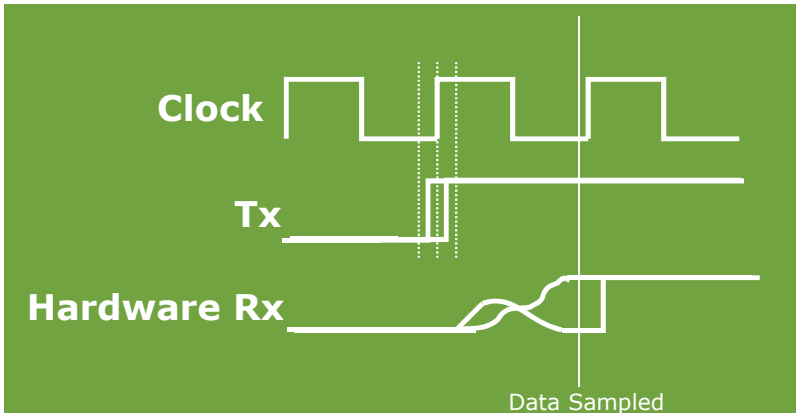
- Asynchronous clock domains
  - Contain registers whose clocks have variable or unpredictable phase relationships vs. other domains
- CDC paths
  - Originate in one clock domain
  - Sampled by register(s) in a different clock domain
- Today's designs can have >10^5 CDC signals!



CDC Paths Cause Metastability

- When CDC signal changes within the setup/hold window of a receiving register
- Receiving register becomes *metastable*
  - Settles to a random value, after unknown amount of time

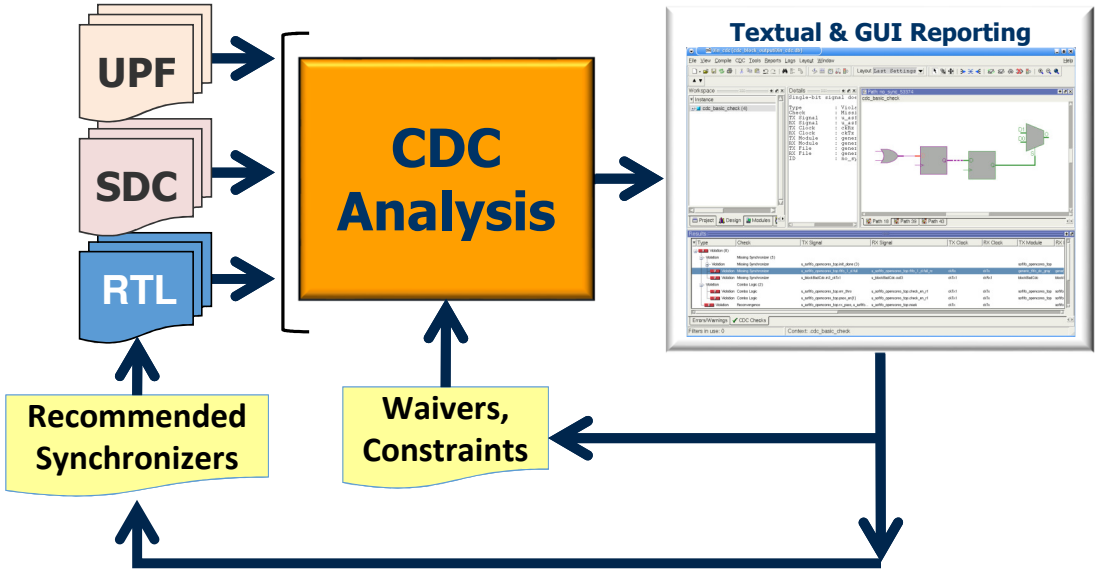




  - Happens even with proper synchronization & protocols
  - Can cause significant functional problems in the design

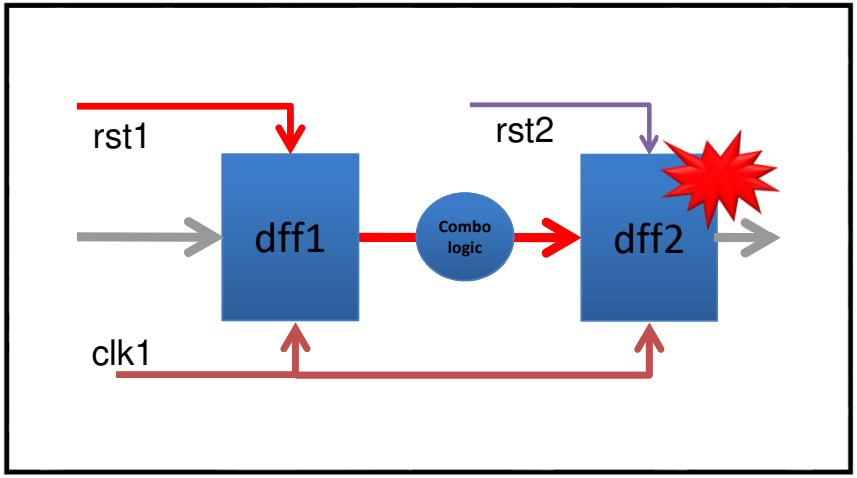
Clock Domain Crossing (CDC) Analysis

CDC verification identifies bugs created by multiple clocks, and suggests circuit corrections



Reset Domain Crossing (RDC)

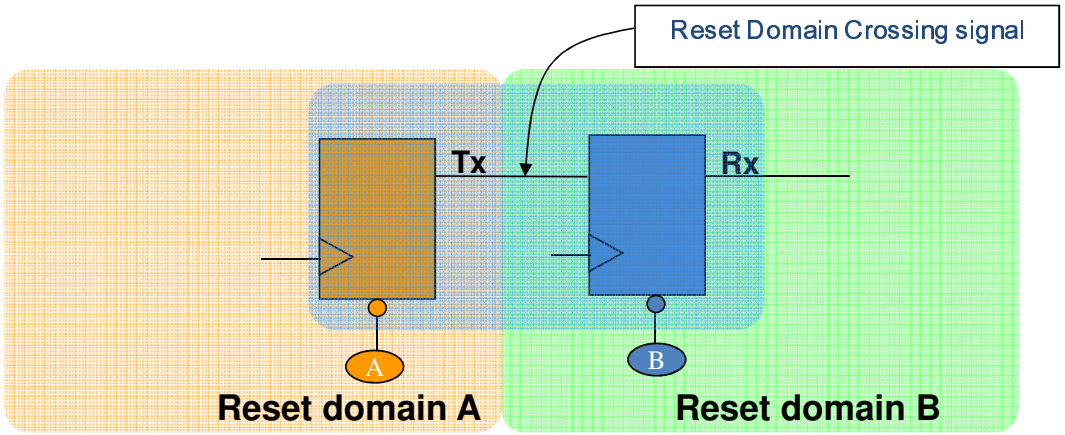
- External/generic IPs used in designs
- Functional domains with independent reset
- Functions requiring async reset
  (e.g. safety, power)

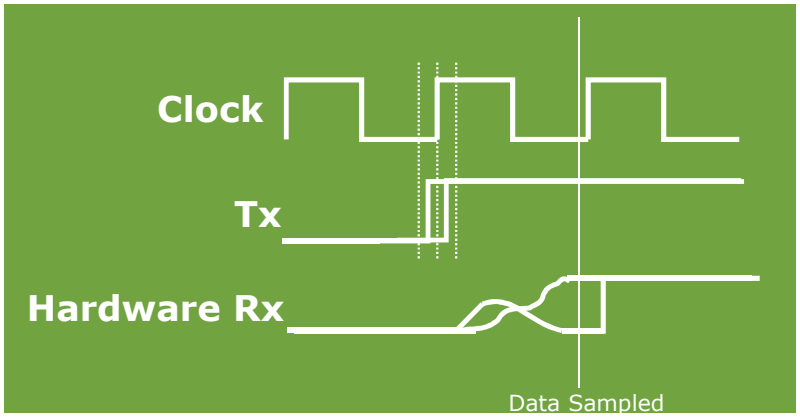


RDC analysis is a similar, but more complex analysis than CDC

RDC Paths Cause Metastability

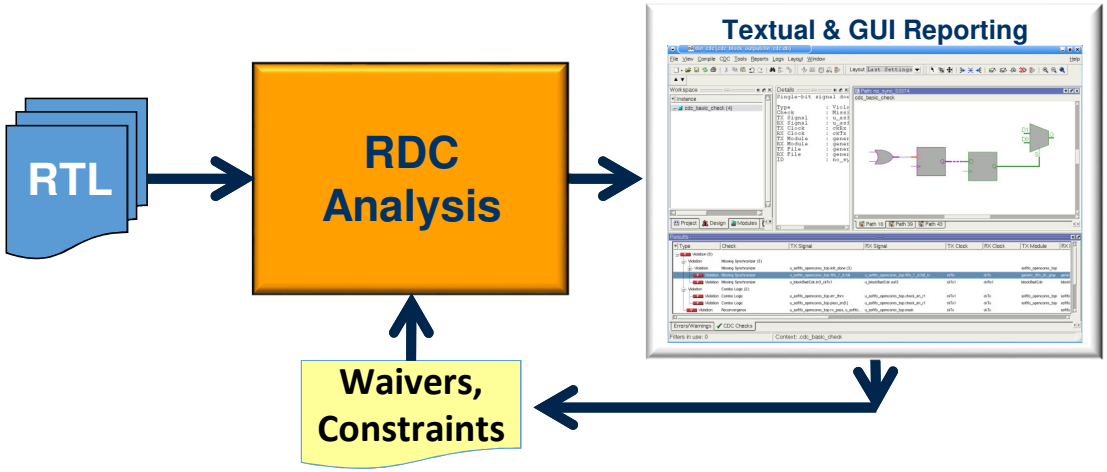
- When RDC signal changes within the setup/hold window of a receiving register
- Receiving register becomes *metastable*
  - Settles to a random value, after unknown amount of time





  - Happens even with proper synchronization & protocols
  - Can cause significant functional problems in the design

RDC Analysis



- Automatically exhaustively identifies all reset signaling issues
- Fully automated analysis: no testbench or knowledge of formal required
- SoC-level scalability: hierarchical approach enables multi-billion gate capacity

Automated FPGA Review Flow Requirements

- Available - tools must be available to the designers at office or home
- Accessible - throughout the design process
- Configurable - to create and enforce project-specific rules and checks, review criteria, and compliance standards
- Identifiable – errors must be easily identified, easily debugged, and provides suggested fixes
- Repeatable - process must produce the consistent results on the same design and configuration
- Auditable - results must be well-documented to allow audit and archival

Results

- Automated FPGA design review flow
- Maintains latest industry standards and best-practices
- Improved performance and productivity
- Enables a well-defined review process

Summary

- FPGA design reviews are critical to mitigate risk
- Manual design reviews are insufficient for complex FPGAs
- Automated review flows address increasing design complexity
  - Increased performance and scalability

Questions