A Closed Loop CDC Verification Methodology

Andrew Cunningham
Agenda

• Introduction
• Standard CDC Methodology
• Standard CDC Results Analysis
• Closed Loop CDC Methodology
• Results
• Conclusions
Introduction Overview

• **DUT**
  - PCIe 2.0 Endpoint
  - 26 clock domains
  - 1.8M standard cells
  - 1 HSIO Hard macro
Introduction Metastability

- aclk is asynchronous to bclk
- aclk is asynchronous to bclk
- aclk
- adat
- bclk
- bdat1
- Only one synchronizing flip-flop
- Data changing
- bclk samples adat while it is changing
- ... but should settle by the next rising edge of bclk
- Clocked signal is initially metastable...
Designers add synchronizer cells to mitigate metastable signals.

Synchronizers are library cells that designed to prevent metastable values being sampled.
Standard CDC Methodology

Standard CDC Flow:

1. Control → Run CDC
2. Run CDC → Validate
3. Validate → Structural Checking
4. Structural Checking → Assertion Generation
5. Assertion Generation → Protocol/Metastability Check
   - No Violations → Design Validated
   - Failed → Analysis
4. Analysis → Analysis
   - False → Waiver
   - True → Real
5. Waiver → Fix Design
6. Fix Design → RTL Design

© Accellera Systems Initiative
CDC Methodology Results

• Categories:
  1. Proven
     • Valid CDC paths which can be mapped to a valid CDC scheme
  2. Cautions
     • The CDC path could potentially cause metastability issues and a protocol checker should be used to verify the interface logic
  3. Violations
     • Signals do not adhere to any predefined CDC scheme and require analysis/debug
CDC Methodology Analysis

CDC Violations and Cautions can be categorized into the following types:

1. Wrong clock specification
2. Quasi-static (stable) signals
3. Dynamic asynchronous Interfaces
4. Non-standard asynchronous design
5. CDC unfriendly designs
CDC Methodology Analysis

![Bar chart showing violation ranges for different types of violations.](chart.png)

- **Violation Types**: Type1, Type2, Type3, Type4, Type5
- **Violation Range**
  - **Upper Bound**: Red bars
  - **Lower Bound**: Blue bars

© Accellera Systems Initiative 2014
Closed Loop CDC Methodology

Closed Loop CDC Flow

- **STA**
- **Control**
- **Run CDC**
- **Validate**
- **Stability Assertions**
- **CDC Violations**
- **No Violations**
- **Design Validated**
- **Sync Cell Count**
- **Regression & Analysis**
- **Waiver**
- **Fix Design**

Flow:
- STA → Control
- Control → Run CDC
- Run CDC → Validate
- Validate → Stability Assertions
- Stability Assertions → CDC Violations
- CDC Violations → No Violations
- No Violations → Design Validated
- Design Validated → Sync Cell Count
- Sync Cell Count → Regression & Analysis
- Regression & Analysis → Waiver
- Waiver → Fix Design
- Fix Design → passed
- passed → Waiver
- failed → Waiver
Closing the Loop

• In order to eliminate all type 1 violations, all IO clocking information is generated from STA sign off tool
  – IO clocking categories:
    • Single Clock Domain
    • Multi Clock Domain
    • Missing clock

• Assertions are generated from CDC violations and simulated
  – Type 2 example assert property ( @(posedge TX_clock)
    !$stable(TX_signal) |-> !config_phase );
  – All violations with passing assertions are waived.
  – If violation assertions fails in simulation, then further debug is required.
Closed Loop Results

Application of Closed Loop updates

<table>
<thead>
<tr>
<th>Flow Run</th>
<th>Violation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run1</td>
<td>Remaining (8000)</td>
</tr>
<tr>
<td></td>
<td>Removed (4000)</td>
</tr>
<tr>
<td>Run2</td>
<td>Remaining (4000)</td>
</tr>
<tr>
<td></td>
<td>Removed (2000)</td>
</tr>
<tr>
<td>Final</td>
<td>Remaining (1000)</td>
</tr>
<tr>
<td></td>
<td>Removed (500)</td>
</tr>
</tbody>
</table>
Conclusions

• Accelerates CDC configuration setup
• Automatic wavering and validation of false violations
• Reduces risk of incorrectly waiving a real issue
• Highlights CDC functional validation holes
• Enables Designer to focus on the real violations
• Faster execution time for overall CDC flow
Questions?