Addressing Asynchronous FIFO Verification Challenge

Anchal Gupta, Ashish Hari, Sulabh Kumar Khare

anchal_gupta@mentor.com, ashish_hari@mentor.com, sulabh-kumar_khare@mentor.com

Design Verification Technologies
Mentor Graphics, A Siemens Business
Introduction

• Asynchronous FIFO (First In First Out) - Advanced synchronization technique used to mitigate metastability risks on Clock Domain Crossing (CDC) paths

• Advantages:
  – Minimal data stability requirement
  – High speed and reliable data transmission

• FIFO verification challenges:
  – Complex data and control logic
  – Implementations vary across designs
  – Prone to structural and functional risks
Risks associated with Asynchronous FIFO

- **Structural Risks**
  - Incorrect clocks used for generating read or write address pointers -> Unexpected behavior
  - Unsynchronized read or write pointers used to determine whether FIFO is full or empty -> Metastability risk

Risk of using unsynchronized write address for generating FIFO empty
Risks associated with Asynchronous FIFO

• Structural correctness is not sufficient to ensure that FIFO will work seamlessly in all corner case situations

• Functional Risks
  – Absence of gray-encoding logic before address synchronization can result in erroneous pointers comparison
  – Data written to the FIFO when it is full
    • Results in potential data loss
  – Data read from FIFO when it is empty
    • Design might reach to a metastable state
Asynchronous FIFO structural verification

• Structural break-down methodology used
  – Complex FIFO structure split into standard sub-structures that can be validated independently

• Standard sub-structures
  – FIFO memory
  – Read and write address pointers
  – Read and write address pointer synchronizers

• Benefits
  – Break-down is independent of FIFO implementation style
  – Helps pin-point the exact logic that needs to be fixed
Asynchronous FIFO structural verification

Break-down of FIFO into sub-structures

FIFO structural verification methodology flowchart
Asynchronous FIFO formal verification

- Ensures FIFO works seamlessly in corner case situations by analyzing data and control logic
- Use formal verification solutions to:
  - Check whether read and write address pointers are gray-encoded
  - Check if data is written into the FIFO when FIFO is full (FIFO overflow)
  - Check if FIFO output is read when FIFO is empty (FIFO underflow)
Case study

• Proposed structural and formal methodology applied on industry microprocessor design, having complex FIFO implementations

• FIFO structural verification
  – Segregated the CDC crossings, synchronized using asynchronous FIFO
  – Identified good and bad FIFO synchronizers

<table>
<thead>
<tr>
<th>FIFO Structural Verification</th>
<th>Number of FIFOs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invalid FIFO read/write address pointers</td>
<td>1</td>
</tr>
<tr>
<td>Missing FIFO read/write address pointer synchronization</td>
<td>2</td>
</tr>
<tr>
<td>FIFO structurally correct</td>
<td>4</td>
</tr>
</tbody>
</table>
Case study

Registers reading FIFO output at a clock different from read clock

Unsynchronized write pointer compared with read pointer to generate FIFO empty signal
Case study

• FIFO formal verification
  – Results showed one FIFO being written in spite of begin full, resulting in data loss
Conclusion

• Generic method to break-down a complex FIFO into standard sub-structures helps pin-point the exact root cause of failure
  – Easier for designer to review and fix the violation

• Formal verification helps in identification of FIFO functional bugs, preventing last minute fixes and silicon re-spins

• Results of CDC verification on several designs using this methodology validates its value proposition
Questions

Finalize slide set with questions slide