Debug Challenges in Low-Power Design and Verification

Durgesh Prasad, Madhur Bhargava, Jitesh Bansal, Chuck Seeley
Mentor Graphics Corp. 8005 SW Boeckman Rd. Wilsonville, OR 97070

Abstract

Various studies have shown that a significant amount of engineering time and effort for a project is typically spent on debug. For a low-power design and verification, these debug challenges are further complicated as a result of the sophisticated power management architectures and techniques that are used. Moreover the traditional debug technology and methods focus on issues found in a design working in always-on mode and fails to address the new and complex power-related issues thereby consuming more engineering time. In this paper, we will provide an in-depth analysis of various debug challenges and problems faced in low-power design and verification. By taking relevant examples we will demonstrate how these issues can be either avoided or solved. We will also highlight some of the common pitfalls that low-power designers can avoid which otherwise can lead to complex low-power issues that are difficult to debug at later stages of the design cycle.

Power UP Failures

Problem: X Values represent floating inputs whose voltage values can cause undesirable design behavior
Solution: Utilize vendors static checks to verify UPF contains the proper isolation/level shifter constructs

Initial block Re-evaluation
Problem: For certain models, such as a ROM memory, initial block may need to be re-executed on power-up after time 0
Solution: Utilize vendors individual solutions to specify which modules or blocks need to have initial blocks re-executed at power up or exclude object from PA semantics

Incorrect or Missing VCT Specification
Problem: HDL/GN D domain pin driven to a logic ‘1’ when connected to UPF GND supply
Solution: Use save_upf command to create interpreted UPF code or use find_objects command to print out expanded TCL variable

Incorrect or Missing RET or incorrect Retention Protocol
Problem: Registers remain X after power-up. Could be missing retention in UPF or non-retention register needing reset on power-up
Solution: Use Waveform compare feature to easily detect simulation differences where X values remain after power-up also enable low power messages and or assertion checks

Determining Sources of Unwanted X Values
Problem: Is X value on a signal due to power domain corrupting
Solution: Use Waveform highlighting to distinguish X value is caused by power domain corruption

Illegal Power States
Problem: How to ensure that unwanted power State and power state transitions don’t occur
Solution: Leverage UPF 2.0_add_power_state and describe_state_transition commands to declare illegal power states and state transitions

Coverage of Power States
Problem: How to ensure that desired power State and power state transitions occur during simulation
Solution: Leverage vendors capabilities in displaying and reporting power state and power state transition coverage data

Achieving PA Coverage Data Closure
Problem: How to ensure coverage closure for power specific items
Solution: Utilize vendors solutions to report, display, and track all coverage data including PA coverage data

UPF 2.0 Migration Issues
Problem: UPF supplies default to OFF state with UPF 2.0
Solution: Utilize UPF package functions to explicitly turn on all necessary supplies

Wildcard expansion issues
Problem: HDL/IP block placed in incorrect power domain
Solution: Use save_upf command to create interpreted UPF code or use find_objects command to print out expanded TCL variable

Incorrect or Missing VCT Specification
Problem: HDL/GN D domain pin driven to a logic ‘1’ when connected to UPF GND supply
Solution: Use save_upf command to create interpreted UPF code or use find_objects command to print out expanded TCL variable

Incorrect or Missing RET or incorrect Retention Protocol
Problem: Registers remain X after power-up. Could be missing retention in UPF or non-retention register needing reset on power-up
Solution: Use Waveform compare feature to easily detect simulation differences where X values remain after power-up also enable low power messages and or assertion checks

Determining Sources of Unwanted X Values
Problem: Is X value on a signal due to power domain corrupting
Solution: Use Waveform highlighting to distinguish X value is caused by power domain corruption

Illegal Power States
Problem: How to ensure that unwanted power State and power state transitions don’t occur
Solution: Leverage UPF 2.0_add_power_state and describe_state_transition commands to declare illegal power states and state transitions

Coverage of Power States
Problem: How to ensure that desired power State and power state transitions occur during simulation
Solution: Leverage vendors capabilities in displaying and reporting power state and power state transition coverage data

Achieving PA Coverage Data Closure
Problem: How to ensure coverage closure for power specific items
Solution: Utilize vendors solutions to report, display, and track all coverage data including PA coverage data

Incorrect or Missing RET or incorrect Retention Protocol
Problem: Registers remain X after power-up. Could be missing retention in UPF or non-retention register needing reset on power-up
Solution: Use Waveform compare feature to easily detect simulation differences where X values remain after power-up also enable low power messages and or assertion checks

Durgesh Prasad, Madhur Bhargava, Jitesh Bansal, Chuck Seeley
Mentor Graphics Corp. 8005 SW Boeckman Rd. Wilsonville, OR 97070

Abstract

Various studies have shown that a significant amount of engineering time and effort for a project is typically spent on debug. For a low-power design and verification, these debug challenges are further complicated as a result of the sophisticated power management architectures and techniques that are used. Moreover the traditional debug technology and methods focus on issues found in a design working in always-on mode and fails to address the new and complex power-related issues thereby consuming more engineering time. In this paper, we will provide an in-depth analysis of various debug challenges and problems faced in low-power design and verification. By taking relevant examples we will demonstrate how these issues can be either avoided or solved. We will also highlight some of the common pitfalls that low-power designers can avoid which otherwise can lead to complex low-power issues that are difficult to debug at later stages of the design cycle.

Power UP Failures

Problem: X Values represent floating inputs whose voltage values can cause undesirable design behavior
Solution: Utilize vendors static checks to verify UPF contains the proper isolation/level shifter constructs

Initial block Re-evaluation
Problem: For certain models, such as a ROM memory, initial block may need to be re-executed on power-up after time 0
Solution: Utilize vendors individual solutions to specify which modules or blocks need to have initial blocks re-executed at power up or exclude object from PA semantics

Incorrect or Missing VCT Specification
Problem: HDL/GN D domain pin driven to a logic ‘1’ when connected to UPF GND supply
Solution: Use save_upf command to create interpreted UPF code or use find_objects command to print out expanded TCL variable

Incorrect or Missing RET or incorrect Retention Protocol
Problem: Registers remain X after power-up. Could be missing retention in UPF or non-retention register needing reset on power-up
Solution: Use Waveform compare feature to easily detect simulation differences where X values remain after power-up also enable low power messages and or assertion checks

Determining Sources of Unwanted X Values
Problem: Is X value on a signal due to power domain corrupting
Solution: Use Waveform highlighting to distinguish X value is caused by power domain corruption

Illegal Power States
Problem: How to ensure that unwanted power State and power state transitions don’t occur
Solution: Leverage UPF 2.0_add_power_state and describe_state_transition commands to declare illegal power states and state transitions

Coverage of Power States
Problem: How to ensure that desired power State and power state transitions occur during simulation
Solution: Leverage vendors capabilities in displaying and reporting power state and power state transition coverage data

Achieving PA Coverage Data Closure
Problem: How to ensure coverage closure for power specific items
Solution: Utilize vendors solutions to report, display, and track all coverage data including PA coverage data

UPF 2.0 Migration Issues
Problem: UPF supplies default to OFF state with UPF 2.0
Solution: Utilize UPF package functions to explicitly turn on all necessary supplies

Wildcard expansion issues
Problem: HDL/IP block placed in incorrect power domain
Solution: Use save_upf command to create interpreted UPF code or use find_objects command to print out expanded TCL variable

Incorrect or Missing VCT Specification
Problem: HDL/GN D domain pin driven to a logic ‘1’ when connected to UPF GND supply
Solution: Use save_upf command to create interpreted UPF code or use find_objects command to print out expanded TCL variable

Incorrect or Missing RET or incorrect Retention Protocol
Problem: Registers remain X after power-up. Could be missing retention in UPF or non-retention register needing reset on power-up
Solution: Use Waveform compare feature to easily detect simulation differences where X values remain after power-up also enable low power messages and or assertion checks

Determining Sources of Unwanted X Values
Problem: Is X value on a signal due to power domain corrupting
Solution: Use Waveform highlighting to distinguish X value is caused by power domain corruption

Illegal Power States
Problem: How to ensure that unwanted power State and power state transitions don’t occur
Solution: Leverage UPF 2.0_add_power_state and describe_state_transition commands to declare illegal power states and state transitions

Coverage of Power States
Problem: How to ensure that desired power State and power state transitions occur during simulation
Solution: Leverage vendors capabilities in displaying and reporting power state and power state transition coverage data

Achieving PA Coverage Data Closure
Problem: How to ensure coverage closure for power specific items
Solution: Utilize vendors solutions to report, display, and track all coverage data including PA coverage data