Low-Power Verification Methodology using UPF functions and Bind checkers
Madhur Bhargava, Durgesh Prasad

INTRODUCTION
- Electronic devices have become complex and energy aware
- Require sophisticated power management architectures and strategies
- If not applied properly, will affect design functionality
- Complex protocols, many power modes need to be verified
- Need for advanced & efficient power aware verification
- Catch low-power bugs at early stages and save design cycles
- Power Intent specification format (UPF) is used to define power management logic - without modifying HDL

Low-Power Verification
- Static Verification
  - Catch all structural errors - correct placement and connections of Pn Cells
- Dynamic Verification
  - Protocol checking - Eg. Check PA Cells are enabled and active at the required time
  - Power Intent checking - Verify power intent specified in UPF against implemented logic.
  - Power Intent coverage - Check if all coverage goals are met for all power aware objects

MOTIVATION FOR METHODOLOGY
- Tool-generated assertions used for low-power verification
  - Some specific scenarios may not be met
  - New protocols may require new assertions not yet supported by any tool
- Custom Assertions & Coverage Items: Club In Check Module and Instantiate into design using UPF command "bind_checker"
  - Low-Power assertions require access of power objects - only present in UPF and not HDL (RTL Stage)
  - Some property checking require design/power control signals spanning across multiple domains.
  - Scope/inputs of checker module defined in UPF - any change in UPF can break these assertions

UPF COMMANDS & REQUIREMENTS IN METHODOLOGY
- find_objects
  - Allow to query the design (HDL) elements.
  - Provides good deal of filtering support to extract relevant elements.
- bind_checker
  - Instantiate checker module (having low-power assertions) into design hierarchy, modifying the design code.
  - Allows one to one port mapping of the checker module to actual power object/control signal.
- query_"
  - Query isolation, query_retention, query_power_domains

AS往訳
- 电源管理を行う電子機器は、複雑化傾向にある一方でエネルギー効率が求められている。そのため、電力情報を取り扱う力のある設計は不可欠である。しかし、不適切に用いると設計機能が損なわれる。また、複雑なプロトコルを使用しているため、多くの電力モードが必要である。そこで、電力情報の認証と効率化が求められている。
- 静的パワーバイアスや動的パワーバイアスの検証を行うため、UPF (Unified Power Format) の仕様が定められている。UPF は、電力管理の仕様をHDL を除く形式で定義し、電力管理を容易にするための仕様である。

METHODOLOGY
- Assertions and covergoers required for low-power verification are packed in a checker module. This module is then instantiated in the design using the UPF command “bind_checker”. Since these SWI and covergoers reside inside checker module, the required design/power signals need to be passed as actual argument to the checker module. This is achieved by using UPF commands query functions and Bind checkers.

Steps for Methodology
- Failin Assertions: Indicate functional issue or a low-power bug
- Coverage Data: Help achieve verification closure

ADVANTAGES OF METHODOLOGY
- Immune to change in UPF
- Highly programmable and easy to use
- Access to all power objects and design signals

EXTENSIONS REQUIRED IN UPF COMMANDS
- query_" command
  - These commands return the object handle as the full hierarchical path of the object referenced from the design top - methodology require the handle of power/control signals needs to be the full hierarchical path.

Some query_" command need to be extended to provide additional information which is not as defined, per the UPF LRM. For example, query_power_domain needs to be extended to provide the primary supply power/ground of the queried power domain.

CASE STUDY
Isolation Protocol Checking: Whenever the driving logic supply is switched off while the receiving logic supply is still ON, an isolation cell is required. One of thing to be verified is that the output port (ap) is closed to golden expected value throughout the duration isolation enable is asserted.

Step 1: Write checker module
- Above check can be expressed in the form of SWA which is written inside a checker module

Step 2: Define interface
- Checker module requires the handle of isolated signals, isolation_enable, clk and parameter values.
- Extract these handles from power architecture (using query_"") and pass these as actual to the formal port names in the checker module.
- Attach the checker module to the design using the bind_checker command.

CONCLUSION
- SystemVerilog assertions and Cover groups
  - used to achieve the verification closure of low-power designs
- Some EDA vendors provide fixed set of low-power assertions and coverage
- Still a need for custom low-power assertions and coverage items
- Suggested methodology using bind_checkers, query commands and find_objects
  - Allows to write some of the very powerful low-power assertion having considerable immunity from any change in the UPF or the design
  - The blind of flexibility the methodology provides in writing assertions and covergroups would be a leap forward in low-power verification.

Note: The purpose of the query_" command in the methodology is to just extract the handles of power/control signals from the power architecture. However any other way apart from query_" commands can also be used in the methodology to extract out the same information.

In fact UPF 2.0 query function definitions were somewhat ambiguous and they have been moved to an appendix in UPF 2.1. The PIBO working group is working on an information model and API that will serve as the basis for a new set of query functions in UPF 3.0.

REFERENCES

Author’s Contact Information

Name: Madhur Bhargava
Email: madhur_bhargava@mentor.com
Company: Mentor Graphics

Name: Durgesh Prasad
Email: durgesh_prasad@mentor.com
Company: Mentor Graphics

© Accella Systems Initiative