

Robust Low Power Verification Strategy for a Complex 3DIC System with Split Power Management Architecture

Atiq Jamadar (<u>atiq.mj@samsung.com</u>) Ayush Agrawal (<u>ayush11.a@samsung.com</u>) Subramanian .R (<u>ramanian.r@samsung.com</u>) Sekhar Dangudubiyyam (<u>sekhar.d@samsung.com</u>) Samsung Semiconductors India Research, Bangalore, India

Abstract- In order to follow Moore's law, chips are manufactured where transistors are integrated into the chip for better performance. But two dimensional scaling will end when gate length and oxide thickness of transistors both reach to their minimum value. Chip scaling improves device performance in terms of gate switching speed but it also has disadvantage of increasing interconnect latency. Three dimensional integration is one of the solution to satisfy Moore's law along with solving the above issue. The basic concept of 3DIC is to integrate more and more transistors in same area while satisfying Moore's Law. Power management is a critical feature of SoCs. SoCs consists of ICs which work on different voltages and different power domains. It is exponentially complex to verify all power domain transition and voltage switching combinations and is critical to the success of the device within verification cycle. Verification time increases exponentially when 3DIC come into picture which has DIE to DIE connected SOCs and each one has its own power domains. This paper addresses this complex problem by developing robust verification environment which accelerate verification of all power and voltage switching within project cycle along with Power State Table(PST) and Low Power Interface(LPI) coverage. This paper also demonstrates hybrid environment strategy which helps to speed up power aware simulation.

I. INTRODUCTION

Large, low-power integrated circuits use advanced technology on-chip power management strategy to control multiple on-chip power domain. Each power domain may support multiple power states representing different voltage/frequency levels at which the domain can function. Transitioning between local power states are achieved through Power Management Unit(PMU). The overall power management state of an integrated circuit is vector of individual local power states of power domain. This strategy tells individual power domains to switch between local power state and thereby control the set of reachable global conditions. The task of designing a power management strategy for some new system-on-chips (SOCs) with many power domains is a significant task, requiring significant effort in design, analysis, and verification.

Power gating is the most effective and widely used power reduction approach. Industry standards have been developed to describe the performance intent of low power designs to simulate power aspects in RTL simulation. However, these features exponentially complicate the verification tasks. It is not uncommon for low power designs to have dozens of power domains and therefore hundreds of power modes. It is prohibitively expensive to verify that a design works in all possible power modes. This complexity exponentially increases for DIE to DIE SOC design where IPs incorporate a low-power design. Fig. 1 shows architecture of one of such a complex 3D IC system where proposed verification strategy is deployed.

In this paper, to overcome this complexity, a robust verification environment is developed that helps to verify all possible power mode transitions, voltage switching, isolation control, and power shutdown conditions within the verification cycle along with code, toggle, Power State Table(PST) and Low Power Interface(LPI) coverage closure.







Fig. 1 3DIC SOC Architecture

II. CHALLENGES IN VERIFYING 3DIC

Below are the challenges faced during the verification of 3DIC

- Controlling power consumption in split power management architecture
- Cross die low power handshake complexity is more compared to single SOC
- Power balls at DUT increases since 3DIC have two SOCs
- Cross Die power domain combinations are more
- Cross Die Power switching combinations are more



II. SPLIT POWER MANAGEMENT ARCHITECTURE

Fig. 2 illustrates split power management architecture. A 3DIC or heterogeneous SOC consists of two SOCs connected through TSVs as shown in figure 1. In general, SOCs have Power Management Unit(PMU) which aims to deliver the maximum designed performance while simultaneously optimizing power consumption for targeted blocks. Since there are two SOCs present its required to have to PMUs for each SOC to control power consumption of respective SOC. There is dependency of one PMU on other. Once the resets of BOT dies are released control would be shifted to TOP die and its PMU releases resets one by one.



Fig. 2 Spin Fower Management Arcintecture

III. CONVENTIONAL SOC VERIFICATION ENVIRONMENT

Fig. 3 illustrates Conventional SOC Verification Environment. SOC consists of different type of blocks working on different power domains. Usually power gating scenarios verification starts once particular datapath of block is successfully verified. Once it's done, conventional approach would be that block owner has to write his own sequence and verify low power datapath of the block. This has to be done for all blocks. Below figures shows conventional method of low power verification targeted blocks. Since there are two SOCs present its required to have to PMUs for each SOC to control power consumption of respective SOC.



IV. ROBUST VERIFICATION ENVIRONMENT

Fig. 4 illustrates Robust verification environment. It consists of one generic sequence which can control power gating, power transitions for all IPs present in both SOCs. It consists of register programming required to disable clock, apply reset of IP and power down the internal component of IP like memories etc. Block owner need to integrate common sequence with his sequence and verify the block's datapath before and after power gating. Below section defines how generic sequence helped in resolving issues and helped in detecting RTL bugs early in verification cycle.





Fig. 3 Robust Verification Environment

1. Low Power Verification

To cover all power domain, robust sequence can be used where all blocks configuration required to be done while power transitioning from PMU instead of creating individual test sequences for each power transitions. When PMU request power down to a certain block, QCH and PCH interface (as shown in figure) starts controlling clocks, resets, isolation cells and power supply of memories in order to achieve low power state of block. Since all blocks have to go through these low power transitions it would be difficult to write test sequence for each block separately specially in case of 3DIC. Therefore, verification of IP's low power state would be convenient, takes less effort and also saves time to write test sequence compared to developing test sequence for each individual block. Design changes are common in whole project cycle and accordingly sequence also need to be changed. As generic sequence is used it's easy to maintain. Changes at one section would be applicable to all.

Fig. 4b illustrates PMU LPI handshake with IPs. Here its show for one SOC only. Other one also have similar type of LPI handshake with IPs. Each IP have its own power domain and power supply. Sometimes IPs also have multiple power domains working on different power supplies. In Fig.4b multi power domain example is taken. BLK-1 has hard macro working on supply2 in power domain A and soft macro working on supply1 in power domain B. Software macro has isolation cells, clock controlling unit(CCU) and memory. Power down process of BLK-1 is explained in Fig. 4a.

Fig. 4a illustrates process of power down of a block. It follows 4 steps.

- Isolation Application → Once power down request from PMU is applied, isolation is applied at the boundary of power domain(PD).
- Clock gating \rightarrow After isolation applied clock are requested to shut off from PMU.
- Memory Power Off → Power switch of memory is turned off. If data of the memory has to retain then
 retention bit is set otherwise data is lost
- Reset Application \rightarrow Applying reset to the block.

It would have simple power down a single block and power switching also. But in case of SOCs where multi blocks are working is complex. Each block has its own one or more power domains and sometime blocks can also work on different voltages. To verify complete SOC along with power gating of single block, combination of other block power gating has to be considered such that to understand effect of power gating one block on other blocks.

To understand complexity let's take an example of a SOC contains 2 blocks and each block working on minimum 2 voltages. Then the combination of blocks to be working SOC would be 4 (ON,ON – ON,OFF – OFF,ON – OFF,OFF). This is with one voltage. For second voltage 4 more combinations. Therefore, complete verification it requires 8 scenarios. Now imagining for 'n' number of blocks working on 'm' voltages resulting in



large number of combinations of power domain switching which is difficult to handle and qualify. Its double when multi die SOCs comes into picture.

Ideally each block owner has to write all sequence for all combination and verify. Instead of this if one common sequence is there then it speeds up the verification of block. Since its controlled by one central PMU before handling working common sequence to block owner it can be run at early stage and find out basic power down handshake bugs at initial stage of verification cycle.



Sequence consist of below power verification scenarios.

• Local Power Gating (LPG)

Below figure illustrates LPG scenario for one block. It means only one block is powered off and powered on and reset of blocks should be ON during this process. It is to check when a block is powered of any X propagation should not happen into ON block from powered off block.



Fig. 5 Local Power Gating Scenario

• All Other Power Gating (AOPG)

Below figure illustrates AOPG scenario for one block. In this scenario Except any one block, all other blocks go into power down mode and come up in power ON state. During this process there should not be any X propagation from powered down blocks to the powered ON block.



Fig. 6 All Other Power Gating Scenario

• SOC Power Gating (SOCPG)

Below figure illustrates SOC Power Gating (SOCPG). In this scenario blocks of SOCs are power gated according to modes set by the designer where each mode defines which blocks are ON and which blocks are OFF. In below case BLK1, BLK3 of TOP die are powered OFF and BLK4 and BLK N of BOT die are powered OFF and rest all are ON. In this scenario which ever blocks are goes into power up down states has to run their blocks functional path and make sure that X propogation from there blocks are not cause any issue in always ON blocks.



Fig. 6 SOC Power Gating Scenario

• SOC Mode Switching(SOCMSW)

Below figure illustrates SOC mode switching of BLK-1. In this scenario out of SOC power modes given by designer one mode is selected and respective blocks functional path has run. During this process SOC modes are switched one to another in a such way that block's whose functional path is running should be ON in those modes.



Fig. 7 SOC Mode switching Scenario



• Common Sequence Implementation

To overcome above explained challenges in verification, sequences are extensively enhanced and modularized based on block requirements. Power down, power up state requirements has been defined as a separate sub-sequence using **`DEFINE Macros**, which in turn becomes part of the main sequence. Fig. 8 illustrates for power down **`POWER_DOWN** macro is defined. For power up **`POWER_UP** macro is defined. In scenario sequence such as LPG, AOPG, SOCPG and SOCMSW a decision macro is used which decides power up/power down of block.

LPG Scenario Implementation



Fig. 8 LPG Scenario Implementation

• APG Scenario Implementation







SOC PG Scenario Implementation



SOC Mode Switch Scenario Implementation







2. Common Checks included in generic sequence

Fig. 13 illustrates block diagram of common checks included in Common power sequence. Sequence has checks like memory retention checks, isolation X-propagation checks and reset checks. These are explained below section.

• Cross Die Isolation X-propagation Checks

Fig. 13 illustrates block diagram of isolation checks implemented with common power down sequence. A python script written to fetch details of input and output signals during runtime of simulation. Once block is powered off, isolations are applied to inputs and outputs of block. On each signal X propagation checks are implemented. Common sequence is coded in a such a way that whenever blocks goes in to power off process and isolations are enabled and X propagations are checked.

• Cross Die Memory Retention Checks

External and internal memory usage is common in SOCs. While blocks are power down its necessary to check retention of data in memories. In common sequence has register configuration which enable and disable retention of memories present in power down block. Memory checks includes sequence to write some random data before power down in memory locations and checks same data is fetched during power down of block where retention is enabled.

Cross Die Reset Checks

Common sequence also includes cross die reset checking where along with the reset of power down block, it also has negative checks which confirms powered up blocks are not reset.



Fig. 13 Common Checks included in common power down sequence

3. Coverage Closure

Of to ensure the functionality of low power design is correct, it is important to make sure that all power states are covered. To verify different power modes supported within each features with minimal changes in test sequence, as an improvement, **\$test\$plusargs API** is widely used as shown in Fig.8. This helps in generating directed scenarios without changing test sequence manually and hence targeted coverage scenario can be achieved without compiling the Test Bench.

Along with this functional coverage of low power interfaces would give more confidence on the design verification especially in case of 3DIC. Left shifting in verification cycle helps to start coverage analysis earlier and helped in finding bugs related to power states and low power interfaces (P-CH Q-CH).



V. RESULTS

Cross die common power sequence is developed for the scenarios like low power gating, all other power gating, SOC power gating and SOC power mode switch. Below are the RTL bugs found at early stage of verification cycle because of which RTL become stable lot more than earlier. It helped to get 2 to 3 weeks more ECO window compared to conventional approach.

RTL bugs found at early stage of verification cycle

- Cross die Low Power Handshake bugs found at early stage which causes delay in verification cycle if found at later stage.
- In a power gating scenario of processor serially (one by one present in top die and bot die) found low power handshake bug where only processor is able turned off but other one was getting stuck in power down process
- Using common sequence, created one scenario where software initiated reset is applied when all blocks are power down except always ON blocks and found a RTL bug where after applying reset complete SOC power up is not happening.
- After power down and power up of block, register writes related to CCU(Clock Controlling Unit) are not happening due to which clock related to block has not set and creating issue in functional path of block.
- Some blocks have internal power modes where top blocks are ON but internal power domains can be turned off as shown in Fig. 4a. The power down inside blocks was not happening properly by keeping other power domains ON in same block
- Found a bug in power up sequence of complete SOC.
- Found UPF isolation issue using isolation checks where wrong isolation value causing complete SOC to turn off when block is turned off

VI. CONCLUSION

This paper presents robust verification environment for verifying all possible power mode transitions, voltage switching, isolation control, and power Includes shutdown conditions within the verification cycle along with functional coverage of low power interface and power mode coverage. This approach is around 40% more efficient than conventional approach and helps to detect crucial bugs at early stage of verification cycle.