2014 DESIGN AND VERIFICATION DVCON CONFERENCE AND EXHIBITION INDIA
Sept 25-26, 2014
Hotel Park Plaza
Bangalore, India

Retention based low power DV challenges in DDR Systems

Subhash Joshi, Qualcomm, Bangalore, India (scjoshi@qti.qulacomm.com)
Sangaiyah, Pandithurai , Qualcomm, Bangalore, India (psangaiy@qti.qualcomm.com)
Halavarthi Math Revana, Siddesh , Qualcomm, Bangalore, India (siddesh@qti.qualcomm.com)

***Abstract*— This paper talks about the challenges & solutions involved in the deployment & verification of retention scheme implemented in a most complex DDR system. Different verification areas such as functional, performance and power are explored which need focus to achieve comprehensive closure of retention based low power verification. Paper also suggests the methodology which may be followed to maintain interchangeable usage of low power techniques across Soc's, inheriting power and functionally verified DDR sub-systems. An example methodology is presented to easily use the retention or save-restore based low power technique interchangeably with minimal effort and time.**

Keywords—Low power DDR Systems; Retention; Verification Challenges;

I. INTRODUCTION

Power Gating or Power Collapse is one of the low power techniques used to operate a block in two power modes: a low power mode and an active mode. The goal is to switch between these modes at appropriate time and in appropriate manner in order to maximize power savings while minimizing the impact on performance. When a sub-system is power collapsed, critical register contents have to be retained in order to bring the system back to its state when powered on. Each power island must be isolated from the rest of the design so that it does not corrupt the downstream logic. The contents of the registers can be saved and restored by software based reads and writes, which is slow with non-deterministic save/restore time due to bus conflicts. The newer and most efficient method of save/restore is by using retention scheme. Retention registers are special low leakage flip-flops used to hold the data of main register of the power gated block. However, power gating with retention achieves faster wakeup times and restoration of state information and, is the most effective technique in reducing leakage power. Power gating controller controls the retention mechanism such as when to save the current contents of the power gating block and when to restore it back.

Verification of designs with power gating is called power-aware design verification and is very challenging due to humongous verification space along with methodology hiccups. UPF defines the retention elements and retention strategies for a system. There were manifold challenges for e.g. getting familiar with power intent of the design, understanding how the tool instruments the retention & isolation strategies, cleverly defining the scope of DV through essential test points, upgrading the checkers to accommodate the design behaviors during post-power collapse, defining power-intent assertion checks and their implementation which can't be covered as part of regular scoreboard checks, defining coverage strategies and hitting them efficiently under extremely tight schedules, dealing with huge simulation time while generating all special sequences from power perspectives are few of the challenges faced during execution of retention based DV for one of the complex DDR system.

Proving integrity of design retention through power sequencing was the biggest challenge against the difficulties as mentioned above. Few techniques such as faster UPF clean-up, Phase #1 and Phase #2 type of DV implementation, improving the simulation time through tool switches and TB enhancements etc. will be discussed in detail to let eco-system aware of the complexities and probable solutions. In summary, Retention based low power DV in DDR System was challenged through various strategies, assertions, tests, simulation optimizations etc. and successfully signed-off.

II. POWER MANAGEMENT TECHNIQUES AND VERIFICATION SCOPE

DDR Systems have been inherent part of all the Soc's. A typical DDR System services multiple processing cores having varying bandwidth and latency requirements. Verifying and debugging DDR System design is challenging because of the speed and complex timing implications posed by memory protocols. Design complexity increases while trying to maximize QoS needs of multiple masters and adhering to DDR memory timings. Apart from maintaining the required data throughput and bandwidth requirement for different on-chip masters the system needs to operate under a power budget. The technology scaling into nanometer regime requires aggressive power management strategies that includes clock-gating, power-gating, multiple-voltages and clock frequencies. These strategies affect functionality if not thoroughly verified which leads to an increasing need for power aware verification.

A. Power Management Techniques

Judicious choice needs to be made before selecting a power aware technique, few of the commonly used technique and criteria to choose between them include:

- Software based SAVE-RESTORE: - Software based save-restore which reads the content of registers before collapsing and restores the content of design registers after collapse. The mentioned techniques include huge software latency where latency is directly proportional to the size of design configuration space. This technique is ideally suited for systems where slow wake-up times could be accommodated. One of the limitation is control path register or decision making registers are required to be part of configuration space.
- Register retention based SAVE-RESTORE: - Register retention based save-restore facilitate quick wake-up times. With few clocks of latency the active contents of registers is restored. The retention flops are special flops used for retaining the content of registers when the design is power collapsed. The retention registers could cover configuration space as well as active decision making space which consists of non-configurable registers.

B. Verification Scope

The low power verification activity begins, when a power architecture is defined for the design. The power architecture details the different power domains (PD), power modes, multiple voltage domains, isolations strategies and power-saving techniques that are deployed in the design. Verification scope increases with each power domain and system needs to be validated under different state combinations of power domains. The verification needs to ensure the power architecture defined is implemented efficiently and correctly. The major areas of verification include:

- Power-domain interactions for e.g. interaction between a collapsible and always-ON domain.
- Ensuring correct isolations are used on power-domain boundaries.
- Design wakes-up in correct state using mentioned techniques and integrity is maintained.
- Ensuring the power intent specifications (UPF) is correct.
- Optimizations possible with retention policies which could save latency and area.
- Scenario Generation, Assertions and Coverage for power aware verification closure.

Power aware Verification and scope needs to be identified and should cover, scenarios focusing on Functional, Performance, Security and Clock gating aspects of design. An example power verification Matrix (PVM) is mentioned in Table1.

Table I. Power Verification Matrix

Matrix	**System Scenarios**				
	Functional	***Performance***	***Security***	***Clock-Gating***	***Multiple PD transitions***
Power	√	√	√	√	√

III. POWER AWARE DV CHALLENGES FOR DDR SYSTEMS

Figure1 represents a simplified memory system connected to multiple masters accessing DDR with different performance and QoS requirements. To facilitate the growing needs of gaming and multimedia application lot of design optimizations are required, introducing enormous complexity. On the other side of interface DDR memory has its own timing implications which needs to be organized, optimized and addressed by protocol engines. Any requirements to performance enhance the protocol engine and make it configurable and programmable across memory technologies adds further complexity to the memory system architecture. Security being inherent part of the eco-system increases further verification scope of the system. The Memory controller scheduled transactions are further converted to protocol format and real world signals by phy which interfaces with the DRAM device. All the features in systems are linked to each other contributing to decision space in form of performance, security or DDR Device state. The performance and power saving techniques further increase the size and complexity of the DDR-system.

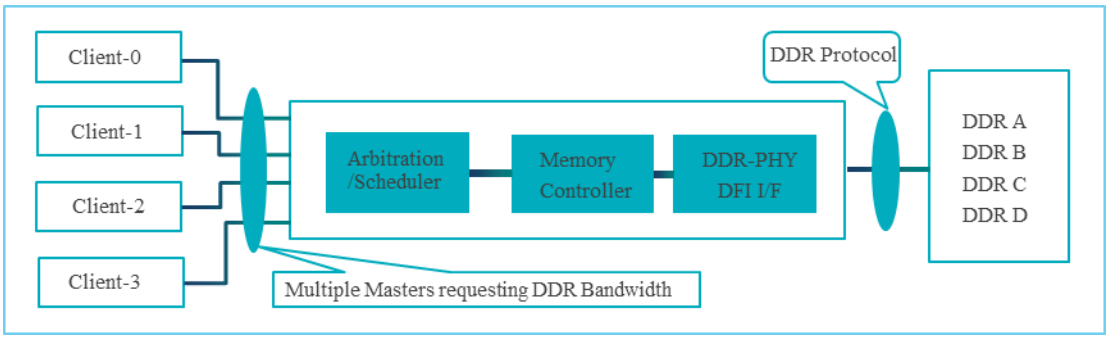

Figure 1. Simplified Memory Sub-System

Power aware verification posed multiple challenges starting from bring-up till closure of activity. The configuration space retention is verified for functionality through register read-write using appropriate verification methodology. The other half of retention space is decision making retention space which requires lot of focus and scenarios to verify the design functionality after power-collapse.

A. Power Aware DV kick-off:

Power aware DV kick-off requires designs to reach certain level of functional maturity. If functional and power verification starts in parallel the verification and debug space grows huge. Issues that occur after power collapse needs to be bifurcated into functional or power, this at times results in complex and time consuming debugs. This requires power intent verification plan development and aligning the power verification activity with functional verification timelines.

B. State Space explosion due to retention

First step in power verifying the DDR-system deploying retention based low power technique is to define the **retention space** with respect to every power domain. The retention space may contain only configurable space of the system or may also have state space (decision making) which is needed to maintain the active state or current state of system for e.g. FSM or DDR-Device status.

The challenges in verification further increase when a freshly build system needs to be power verified at DV scope, when the retention list is new and has to be verified from scratch.

- **Issue is Retention or Non-Retention:-** Whether a flop needs to be retention or non –retention needs in-depth micro-architectural understanding and depth increases with the size of system under consideration. Issues could arise by keeping a non- retention flop into retention list as well as retention flop getting excluded from retention list. The verification and debug becomes extremely challenging when a missing retention flop needs to be identified from entire system-space, which may be present deep into microarchitecture. Or to identify a flop which should not be present in the retention list. These issues may or may not result into functional failures.

- Bringing-up of first power aware simulation pass could result into complex debugs due to above mentioned issue-sources. Missing Retention Flop: - In a DDR System, ddr-device type is used by

multiple modules which could have internal logic to store the device type information on certain conditions, every such module need the internal register to retain the device type information for correct operation. Missing Non-implied periodic commands as an examples could never be discovered without robust verification checks. In-correct Retention Flop: - There could be certain logic in design which may require registers to capture reset pulse information, such flops kept in retention could never capture reset pulse information, due to power collapse sequence. Contents of flops are saved prior to reset and same is restored post reset.

- Silent Bugs – Performance issues or unnoticeable bugs allow verification scenarios to pass functionally but result in performance holes. These issues are difficult to find and result in change of the system performance after power collapse. The source of these issues could be retention registers which belong to non-configurable space of design and are missed out in the retention list. Robust power verification of system requires entire performance test-plan to be executed in power aware environment. Cross-coupling of performance and power scenarios may also result in verification effort due to various reasons such as Directed test-cases, assertions and performance checks not accounting for power –collapse.
- Example of performance issues could be, two reference retained value Ret_A, Ret_B from configuration space are compared to create a pointer, which could point to the optimal settings based on certain conditions. If the pointer existing on decision/state-space is not retained the system may be working under non-optimal settings after power collapse. Another example could be suppose the design register to be retained Reg_A, is updated under certain conditions (Cond_A) and is used to apply performance settings. Assume the system state is Active and Reg_A has active contents pointing to performance setting Perf_A. When system comes out of power collapse and contents are restored. Reg_A may not get updated as condition Cond_A is not fulfilled after power collapse. Causing system to operate under incorrect performance settings till Cond_A is satisfied.
- Security: Retention verification should also ensure that access policies and protections in the system is verified thoroughly post collapse.
- Clock-Gating/Dynamic Power: All the clock gating schemes and design elements needs to be checked for correct behavior during and after collapse. For e.g. the activity based clock gating scheme deployed in most of the systems may become non-functional due to retention misses. For e.g. free running clocks connected to DDR is only power issue and may not result in any functional failure.
- Live Lock / MC to PHY Interface: - Power collapse scenario in DDR-System also requires software sequences to control handshakes. There could be issues when dual handshake mechanisms are used. Scenarios where system goes into power collapse with an active handshake.
- Software limitations and USECASEs: - Power collapse verification at memory sub-system could produce many interesting scenarios, which may not be a System use-case. This also helps in identifying the software limitation upfront when non-use case scenarios occur during debugging and development stages.
- Aggressive Timelines and Performance fixes: - Power verification of functional features starts after certain functional maturity of design. After functional features are stabilized in power aware verification, Performance verification of system in power aware system starts. There could be performance issues present while doing power aware verification. The fix for such issues could be difficult and depends on the product cycle time. Timelines define if a hardware fix could be accommodated or software workarounds could be deployed.

C. Configuration based save-restore

The Configuration space based save-restore scheme produce huge software latencies to bring system to active state. The size of memory required to save the active state information also increases with increase in configuration space or size of system. This technique requires the dependencies between register writes to be maintained while restoring, issues could arise if two dependent registers are written in incorrect order.

A thorough verification needs to be carried out by coupling power collapse scenarios with performance, security, functional and power aspects of design. Any failures in these aspects of design would require registers or flops to be mapped to configuration space of design.

D. Isolation

To reduce leakage current, Power-Ground supplies to the collapsible domain are disconnected. This causes the ports driven from collapsible domain to float. The output ports of the collapsible domain may be connected to an active domain, during collapse this may cause floating ports to corrupt the logic present in active domain. To avoid this collapsible domains are isolated from active domains using isolation cells. The isolation value needs to be verified as improper isolation may cause the active domain to reach un-intended states.

Missing isolation gets verified through static checks, whereas dynamic checks are used to verify the isolation behavior. Verification is done through an assertion package which could be automatically bound to the ports of collapsible domain. Following category of assertions could be used to ensure quick identification of incorrect isolations

1) Isolation values on ports should match the reset value.
2) Ports should not toggle during power collapse or when isolation enable signals are asserted high.
3) No X or Z propagation on ports.

E. Coverage convergence and sign-off

The power aware verification environment leverages functional verification infrastructure and requires power related modifications for accurate and robust analysis. The Assertions coded for functional verification may require modifications under different cases

- Case 1: Scenarios if assertion variables are initialized using initial blocks the basic example could be free running counter. Power aware simulation requires these variables to be initialized on resets.
- Case 2: Assertions need to account for signals going x during power collapse time.

Functional coverage model for system could be leveraged from functional verification environment. Capping bins needs to be added, to ensure coverage bins are hit only when functionality or features are active post collapse.

Verification closure for retention based technique, requires functional coverage matrices to be developed for all the registers/flops mentioned in UPF. An automated approach could be adopted by binding the retention list to the power sequencing model and creating automatic functional coverage bins for retention of 1 or a 0 value for a flop. The analysis of flops missing retention is difficult as it requires entire system understanding. It may become more difficult if the flop names doesn't correlate to any of the design feature or register directly. Closure requires every single retention flop to be analyzed for functional coverage.

Coverage closure on Power aware verification may also be used to scrutinize the sub-system with 100k flops in retention and identifying different category of flops which may not require retention. This provides an area intensive feedback from retention perspective. Category of flops which may not require retention include (a) flops/registers which are not active during power-collapse and (b) A 32-bit register which may never toggle few of the bits during power collapse size.

IV. POWER AWARE DV SOLUTIONS FOR DDR SYSTEMS

Figure2, represents the strategies adopted to tackle DV-challenges faced while using retention based low-power technique.

Phase I: - The configuration space retention is verified for functionality through register read-write (effective patterns) using appropriate verification methodology.

Phase-II: - Verification of decision making retention space is challenged by developing comprehensive test-plan. Robust power test-plan development requires lot of focus and scenarios to ensure retention list is complete

and all necessary retention flops are identified. All the design-features need to be regress tested under power aware environment to ensure design functionality after power collapse.

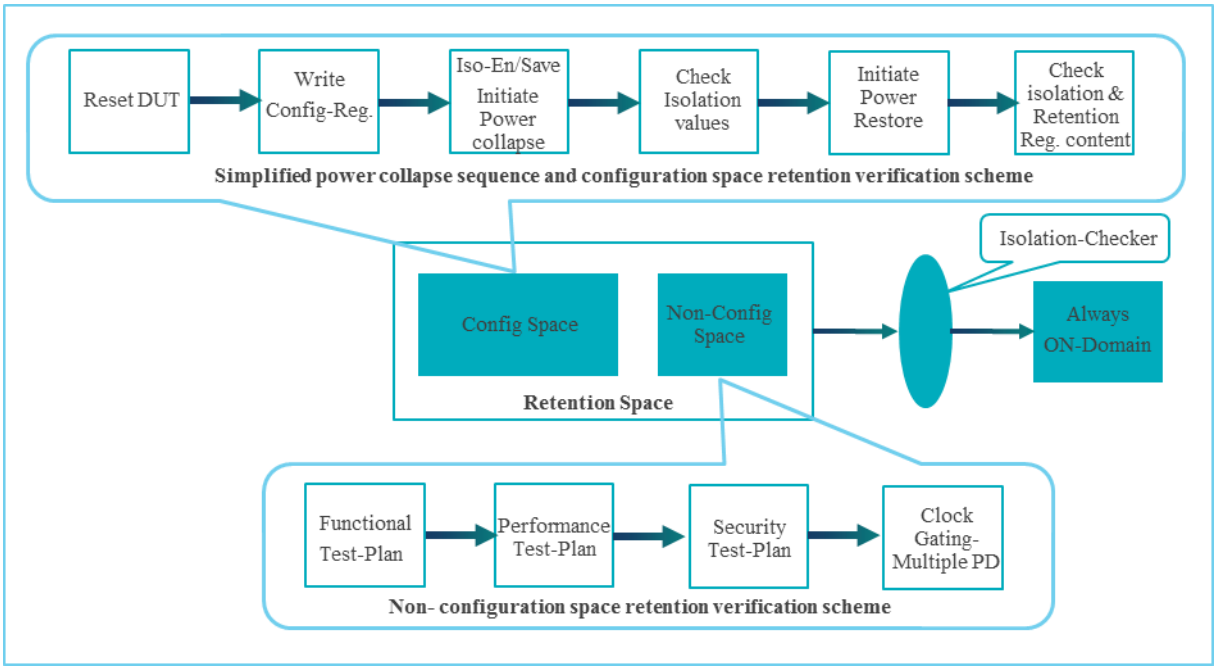

Figure 2. Power Aware DV Solution for retention based verification

A. Comprehensive testplan covering functional, power and PERF aspects

DDR systems are one of the control intensive designs, which may have lot of decision making registers not mapped to configuration space. Extensive test-planning is required when such systems deploy retention based techniques or when the retention list is new. The power verification test-plan must cover power verification matrix (PVM). The DDR-system power test plan must include:

- All functional test-points/scenarios added to power-aware test-plan. The scenario should ensure the feature is enabled before power-collapse and is checked for proper operation post collapse. One of the example is DRAM-De-rating feature.
- All performance test-points/scenarios added to power-aware test-plan. The scenario should ensure performance number measured before power-collapse and after power-collapse co-relate. The scenarios should detect temporary performance holes created by retention misses. These kind of scenarios typically look like [Power-up -> Event (E_A) -> apply performance setting (PERF_A) -> power collapse -> performance setting modified (PERF_Changed) -> Event (E_A) -> regained correct performance settings].
- All security test-points/scenarios added to power-aware test-plan.
- Test-plan should ensure scenarios with Clock-gating enabled and verified after power-collapse. The clock gating needs to be verified across entire system-space. Un-gated clocks may not result into any functional errors. Drams provide features for stopping clocks after entering to power-down modes. Memory simulation models may not result into any violations and power-aware checks may need to be added/modified.
- Power Aware Test-Plan must include configuration space retention-verification using effective read-write patterns.

- MISC:- Power Aware test-plan should include regular power intent checks For.eg. Multiple PD state transitions/ combinations, isolation checks and level shifter operations.

B. Planning and Execution Strategies

Tight timelines require accurate planning and execution strategies to mitigate the complexity and effort involved in fixing a late bug, the entire power-aware verification activity is accomplished in different phases as follows:

- Phase I [Bring-Up]:- Power aware verification starts off with cleaning-up UPF, providing behavioral power models and bringing-up of CAD-tool environments for power-intent verification. Configuration retention space verification flushes most of the retention issues. This also requires knowledge of non-retainable configuration space. Bring-up test may also cover entire system data-path flushing any basic issue with retention list and isolation values.
- Phase II [Directed]:- Once the bring-up is completed, all the design features need to be enabled in the power aware simulations and thorough scrutiny is required. The effort also extends towards modifying assertions to account for power-aware scenarios. Generating power-coverage bins by capping power-aware events on top of functional coverage model.
- Phase III [Random]:- After dedicated feature verification is completed system needs to be enabled for all the features in simulations and should go through multiple collapses to regress power integrity of design. Random scenarios need to be created which verifies design through multiple power collapse, switching to different power modes, switching on and off different system features while sweeping through multiple frequency of operations. Randomization of DDR aware traffic and design operation under different DDR frequencies needs to be analyzed.
- Prioritization of Functionality over performance: - Tight timelines require functional scenarios to be qualified before executing performance test-plan in power aware environment.

DV infrastructure for power verification may need to ensure all the features are power verified and entire system use-case scenarios are enabled throughout regressions, generating corner scenarios for cross-coupled features.

C. Selection of appropriate methodologies/tool flows and automations for faster convergence

In scenarios when a functionally verified system is used in another SoC, which may be implementing a different save-restore scheme then verified in previous version. Methodology needs to be followed to bring the configuration based save-restore technique in sync with retention based save-restore technique. The end goal of both the techniques is to ensure that design integrity in terms of performance, security, power and functional features is maintained after deploying any of the LP technique.

Retention based save-restore verification may identify many flops getting added to retention list from decision making space or non-configuration space. The updated retention flops list may allow design to function properly. When the same design is ported to a different SoC and the configuration based save-restore techniques is used it may behave incorrectly for all the issues identified in retention based save-restore technique. To ensure the sync between both the techniques all the missing retention flops/regs needs to be connected to configuration space. This would ensure proper functioning of design irrespective of deployed low-power technique.

V. GUIDELINES/TIPS

DV Guidelines which may be used to ensure quality and robust verification include:

- Verification environment development must ensure minimum usage of directed tests. The practice of verifying design feature by a dedicated test may mask interesting scenarios. Verification environment must be easily configurable to enable and disable any feature which may allow corner scenarios to be created.

- Assertions coded during functional verification must be written assuming power collapse scenarios.
- Functional coverage models must take into account power collapse events to ensure correct hit of functional coverage bins during power aware simulations. The coverage bins must be hit only after power collapse to ensure features are verified accurately after system is restored.

VI. CONCLUSION

Power aware verification adds new dimension to verification scope, robust verification is ensured by developing power aware assertion and functional coverage models for sign-off and closure. The power aware plan needs to intersect with Performance, security and all features of design. Methodologies needs to be followed to maintain consistency between different low-power techniques for ease of use and quick turnaround time.

REFERENCES

[1] IEEE Std 1801™-2009 for Design and Verification of Low Power Integrated Circuits. IEEE Computer Society, 27 March 2009.

[2] Amit Srivastava, Rudra Mukherjee, Erich Marschner, Chuck Seeley, Mentor Graphics, Sorin Dobre, Qualcomm: "Low Power SoC Verification: IP Reuse and Hierarchical Composition using UPF", DVCon 2012.

[3] Stephen Bailey,Gabriel Chidolue,Allan Crone,Mentor Graphics "Low Power Design and Verification Techniques".

[4] Noah Bamford, Freescale Semiconductor, Saji George, Freescale Semiconductor, Milind Padhye, Freescale Semiconductor "Verification of Low Power Designs using CPF".