Achieving First-Time Success with a UPF-based Low Power Verification Flow

Kjeld Svendsen
Applied Micro
215 Moffett Park Dr.
Sunnyvale, CA 94089
+1 408 542 8302
Ksvendsen@apm.com

Chuck Seeley
Mentor Graphics Corporation
Wilsonville, OR
+1-503-685-0816
Chuck_seeley@mentor.com

Erich Marschner
Mentor Graphics Corporation
Ellicott City, MD
+1-410-750-6995
Erich_marschner@mentor.com

ABSTRACT

Minimizing power consumption has become a critical requirement in today's designs. Active power management required to minimize power consumption creates additional challenges for functional verification. IEEE Std 1801™-2009 [1] defines the Unified Power Format (UPF), which enables visualization and early verification of the behavior of a design under active power management during RTL simulation. This paper describes a UPF-based low power verification methodology used by Applied Micro Circuits (APM) for verification of a low power design, including the process used for verification planning, tool flow, and methods used to track progress toward coverage closure.

Keywords

Multi-processor system, Low Power, Functional Verification, UPF Unified Power Format, IEEE 1801

1. INTRODUCTION

Power management has become a critical concern in the design of electronic systems, especially for those intended for very low power applications. Larger (e.g., multiprocessor) designs may contain so much logic that it is impossible to power up all parts of the chip at the same time. Power management within a chip has become mandatory for designs such as these.

Minimizing power consumption and consequent heat generation, and for portable systems, maximizing battery life, are key requirements for successful products today. Minimizing power consumption through clock gating was sufficient for older process technologies, but more recently, with the continued advance to smaller and smaller process nodes, static leakage has become a major issue as well, representing as much as two thirds of the power consumption of modern designs. The need to minimize static leakage has led to new power reduction techniques [2] such as power gating, biasing, and multi-voltage supplies, which in turn require power management architectures that enable and support the use of these techniques, as well as both hardware and software control logic necessary to initiate and mediate transitions among the various power states of a system.

Various power management techniques are in common use today. Clock gating disables the clock of an unused device, to eliminate dynamic power consumption by the clock tree. Power gating disconnects a device from its power supply during standby mode, to eliminate static leakage. Body biasing changes the threshold voltage to reduce leakage current at the expense of slower switching times. Voltage scaling changes the voltage and clock frequency to minimize static leakage while still meeting performance requirements. Multiple voltages can be used for different parts of a system that have different performance requirements. One or more of these techniques may be used to minimize power consumption in a design.

Power management must work correctly while at the same time enabling the design itself to function correctly. Changes from one power state to another involve a sequence of operations that must be orchestrated correctly to ensure that neither logical nor electrical problems occur. The fact that portions of a chip may be powered down or in a low power state at any given time requires logic to isolate those portions from other parts that are operating normally. Interactions between power domains operating at different voltage levels requires level-shifting logic to ensure that logic values are correctly transmitted and received. State retention logic may be required to preserve key data across power-down periods or to enable a given power domain to power up quickly without lengthy reinitialization.

Verifying active power management at the IP block level involves both verifying the power management architecture – the structures that provide control over power gating, mediate interactions between power domains, and enable state retention – and verifying the power management behavior – the operation of the power management architecture together with the design, given appropriate sequences of control inputs. At the system level, power management includes verifying the correct sequencing of power management control signals as well as thorough verification of all the system power states and transitions among them.

2. THE APM DESIGN EXPERIENCE

APM has developed numerous designs involving power islands for power management. Previous designs employed manual insertion of technology specific isolation buffers in the RTL design, which caused inefficiencies in design technology portability and in particular verification and confidence hereof. The verification issue was that internal state of powered down blocks did not go 'X', but retained any acquired previous logic state.

This had two consequences. Firstly any powered down block that drove logic outside the block would not drive an X potentially causing missing or incorrectly isolated logic from being detected, and missing any consequences of nets being driven from powered down block, which could cause complete malfunction in silicon. Secondly upon repowering of the powered down block(s), one could not tell if the blocks would correctly repower with uninitialized state or not, since the internal state would not be driven to X.

Based on these prior experiences the APM86290 design [3] team recognized early on in the definition phase as requiring more sophisticated verification capabilities due its more elaborate power management mechanisms. The APM86290 is a high complexity SOC with dual IBM46x PowerPC processors organized in a SMP configuration for cache coherent operation between processors and I/O and has numerous high-speed interfaces on-chip combined with several power domains to permit a power operation range of 10uW to 6W.

To facilitate power management the SOC subsystem design features the Scalable Lightweight Intelligent Management processor or SLIMpro^{TM}, a dedicated very low power micro-controller managing power and reset sequencing, so individual processors, both processors and individual I/O subsystems can be power sequenced, permitting configuration management for the most power efficient yet high-performance throughput system based on application requirements.

Individual power islands on the chip are powered by external voltage regulators, as the high-end power consumption of the chip doesn't permit on-die power switches.

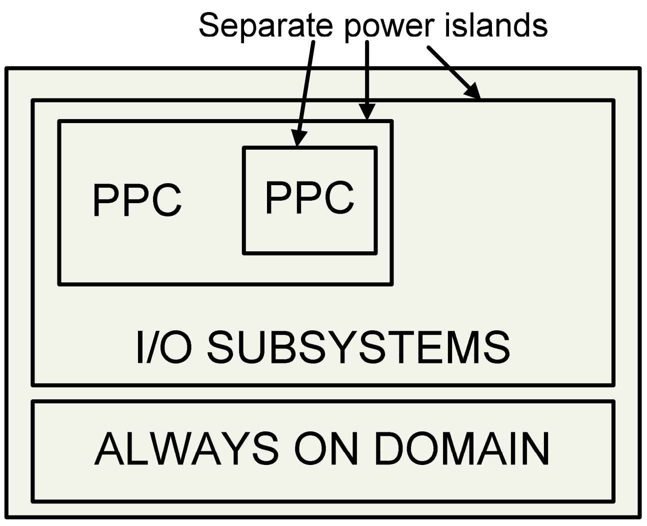

Figure 1. APM86290 Power islands

APM decided to embrace the industry standard UPF-based methodology as this was readily supported by their Mentor Questa environment and had a convenient design-tool flow path to the backend synthesis and P&R tools.

It was nonetheless recognized early on that the power management architecture and related functionality needed to be clearly defined and specified up-front, and not considered as a quick do-it-later add-on feature. The specification involved both hardware and software permitting hardware-software co-development and verification.

2.1. SPECIFICATIONS AND TESTPLANS

The power architecture specification detailed all power regions, power configurations, specific power goals on a per configuration basis and system software interface and sequencing procedures. I/O devices and system aspects to power sequencing were also detailed to assure end-system compliance.

The architecture specification was further very detailed, describing strict methods with exact timing diagrams and sequences limited to specific behaviors to minimize error-proneness due to elusive corner-cases, engineering miscommunications and to minimize the verification effort. Of particular concern was the sequencing of power, isolation and reset going into power-off and into power-on. For the power-on case, isolation should be set and reset should be applied and held for a period of time until power is stable and the internal state properly reset. Then isolation can be removed, and then again after some time reset de-asserted. Figure 2 shows a waveform representation of the power on-off flows.

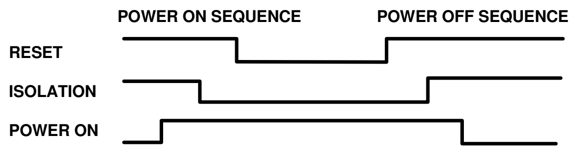

Figure 2. Power Sequencing

On power-off of a processor, the processor first needs to be completely quiescent, which was achieved through an internal processor mechanism. It should be noted that quiescent in the context of MP systems involves a multitude of other aspects than just signal quiescence, e.g., all caches must be flushed, all in-flight transactions completed, snoop queues empty, etc., all which are beyond the scope of this paper. The HALT instruction would thus indicate through a signal interface that the processor was ready for power down. Then reset could be asserted followed by isolation and then power off.

It should be noted that the processors did not have any internal configuration state that required state retention. Such state was kept on the outside the processors' power domains in an always-on power domain.

The architecture specification permitted a detailed RTL development plan to be written, starting with a clear-cut design specification to be developed outlining which blocks were impacted by the power sequencing and how the different blocks would interact as a consequence.

The testbench and test development also benefited from the detailed upfront architecture specification as the test complex could be developed with the power sequencing requirement in mind. In particular at the subsystem level, e.g., for the CPU complex the encapsulating testbench was developed with a pseudo power sequence controller emulating the SOC level SLIMpro^{TM} processor, permitting testing with the exact sequencing and timing in a simple manner.

Testplans were developed, which specified all the tests required for all the power configurations, and coverage objects for all the power sequences and power configurations.

A key decision was made to include actual system software in the verification plan to assure software-hardware integration correctness on a pre-tape-out basis. It was naturally felt this type of verification would be very time-consuming, but the runtime was found acceptable as it was estimated possible on an overnight run basis on high-end servers and minimal iterations would be expected given the pre-verification should weed out all major hardware issues, and thus the runs should work with no hiccups except for software issues. But the potential time cost was considered acceptable given the alternative time and monetary cost of malfunctioning silicon.

2.2. RTL ORGANIZATION

As RTL development was gated by the architectural specification, it was possible to properly plan the RTL organization and design effort with the power architecture fully integrated upfront.
The RTL was accordingly partitioned along power domain borders, greatly simplifying the UPF specification, verification and debug and backend work. It is possible to specify power domains in UPF with logic scattered across multiple blocks, but this was considered error-prone and was thus deliberately avoided.

The RTL was coded with the needed power-on reset and power sequencing signals which would then be connected automatically by the back-end tools thru the UPF specification.

OVL assertions were added to the RTL code to cover incorrect and/or unexpected power sequencing. This allowed for quick determination of miscommunications and misunderstandings between design, verification and software development, and trivialized potential disasters by pin-pointing critical issue on a very immediate basis. The UPF code itself also includes a behavioral description that permits checking of power sequence violations.

2.3. TESTS, TOOLS AND FLOWS

An important aspect of the overall power verification effort was assurance of full design tool flow integration from specification to the tape-out database. Mentor's Questa power aware simulator readily supported the RTL to UPF integration in an easy to use manner. The compiler/ simulator can include a UPF file specification on the command-line, which permits fast integration.

Using UPF enabled technology independent verification, where the library isolation cells need not be specified yet. In contrast, the target library isolation cells are required for a CPF specification. Thus with the UPF format, the power domain aspects of the design can be verified before and without a cell library available.

The testbench code used the standard UPF interface subroutines to control the power sequencing. The routines were readily made available through a library import statement in the testbench:

*Import UPF::**

And routines herein were then called e.g.:

supply_off("streak_tb/streak/pVDD1");
supply_on("streak_tb/streak/pVDD1",0.99);

The signals specified in the function calls are correlated with the RTL through the UPF file.

In the second *supply_on* statement, the 0.99 indicates the power on voltage and would be flagged if the voltage was specified differently in the testbench than in the UPF file. This is a feature provided by the UPF format for systems that use multiple voltage levels, e.g., Dynamic Voltage Scaling (DVS), but was redundant for this particular application.

Verification using the simulator quickly identified issues with the UPF specification, RTL coding, and general functionality. Coverage was collected as per the specified coverage points to assure all ends of the design were tested. This was readily available through the Mentor toolset.

Running actual system software was shrink-wrapped to the power sequence subroutines, which simplified the verification, shortened the runtime, but still provided the looked-for software-hardware integration assurance.

The back-end tools posed an initial challenge as they used the CPF format which is not compatible with UPF. The design team was aware of this limitation upfront, but as Questa supports UPF and early verification could commence before cell library availability, UPF was used. Furthermore, using Conformal LEC, a CPF file could be written which, when read in with the RTL, could generate a UPF file after elaboration. This permitted connecting the front-end tools to the back-end tools using a single source CPF file. The flow is shown in Figure 3.

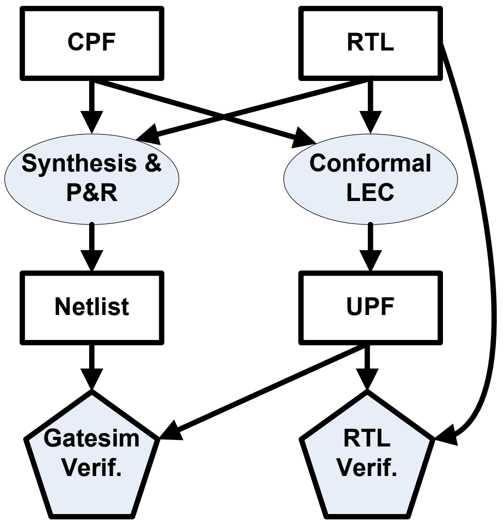

Figure 3. Power Verification Flow

Naturally this meant an initial UPF file had to be written for early verification, then a CPF file written for the back-end tools, and then the verification rerun with the UPF file generated from the CPF file. However, at the time the design went to the back end, the power tests were completed and the verification rerun a minor affair.

The CPF-UPF conversion was admittedly not perfect and did require a minor tweak in the naming of power nodes. This was not considered reason for any concern, although it would have been nice to have better translation correspondence from the naming conventions used in the CPF file to the generated UPF file.

With a netlist, gate simulations could also be run, and though slow, this permitted assurance that the netlist worked the same as the RTL. The Mentor Questa simulator further provided the same corruption of powered down blocks' state in the netlist as for the RTL, which provided good verification confidence. Gate simulations were run both in Zero delay and with SDF, with the latter permitting observation and resolution of potential timing troubles, which could occur due to delays incurred due to the isolation buffer insertions, including buffer delays and turn-on/turn-off delays.

2.4. UPF CODE

For reference the UPF code for the dual PowerPC processor (Tiger) complex (Streak) is listed below to exemplify the actual design.
Key elements are the power domain structure specification which specifies the two power domains PD1 and PD2. There is a processor

in each domain, but only the processor in power domain PD2 is specified as able to be powered down.

Since the processors are designed in logical blocks the power domains are easily specified, e.g., tiger_1/tgr. The individual power ports are specified for connections to the power isolation controls. Isolation is specified to define the operation of the power isolation buffers. But the isolation buffers do not need to be specified as the back-end tools will pick these automatically. This is handy for a technology-independent design. Isolation can be specified on inputs and outputs, but output isolation was found to be sufficient.

Some care was required with the isolation_power_net specification in the UPF set_isolation command. This port needs to be connected to the outside power domain net, because if it is connected to the same VDD as the power domain that is powered down, the isolation buffer will go X when the power is removed.

As a side note, it was found that the isolation buffers' leakage current can add somewhat to the power consumption. Also since the isolation buffers are virtually instantiated thru the UPF simulator interface they are not viewable in the signal trace tool (neither in Questa's GUI nor in Debussy) which was found to be a slight nuisance. The isolation control identified the signal used to control the isolation buffer, and its active level, i.e., the level that causes the isolation buffers to isolate. Getting the hierarchical signal-path right proved a little tricky as the testbench and top-level RTL paths need not be specified, but after some experimentation we prevailed.

Lastly the power states and transitions are specified. This is handy for verification as incorrect state and/or sequencing would be flagged permitting a quick identification of otherwise easily overlooked violations.

```
# Define the top
#====================================================
set_design_top streak

# Set up logic power domain structure
#====================================================
create_power_domain PD1
create_power_domain PD2 -elements {tiger_1/tgr}

# Create power I/O
#====================================================
create_supply_port pVDD -direction in
create_supply_port pVSS -direction in
create_supply_net nVDD -domain PD1
create_supply_net nVSS -domain PD1
connect_supply_net nVSS -ports { pVSS }
connect_supply_net nVDD -ports { pVDD }

create_supply_port pVDD1 -direction in
create_supply_net nVDD1 -domain PD2
create_supply_net nVDD -domain PD2 -reuse
create_supply_net nVSS -domain PD2 -reuse
connect_supply_net nVDD1 -ports { pVDD1 }

set_domain_supply_net PD1 -primary_power_net nVDD
-primary_ground_net nVSS
set_domain_supply_net PD2 -primary_power_net nVDD1
-primary_ground_net nVSS

# Set isolation
#====================================================
set_isolation ISO_PD2 -domain PD2 -isolation_power_net nVDD
-isolation_ground_net nVSS -elements { tiger_1 } -clamp_value 0
-applies_to outputs

# Set isolation control
#====================================================
set_isolation_control ISO_PD2 -domain PD2 -isolation_signal
streak_cpm/cpm_isolatecpu1 -isolation_sense high

# Define static behavior of all power domains
#====================================================
add_port_state pVDD -state {ON 0.99}
add_port_state pVDD1 -state {ON 0.99} -state {OFF off}
create_pst PST -supplies {pVDD pVDD1}
add_pst_state PM1 -pst PST -state {ON ON}
add_pst_state PM2 -pst PST -state {ON OFF}
```

3. CONCLUSIONS AND FURTHER DEVELOPMENTS

The UPF flow usage was found to be highly productive and provided a high level of confidence in the final design, especially compared to prior design experiences. Actual silicon validated the value of the design and verification process by providing fully functional first silicon for the APM86290 processor.

The effort involved to add the UPF portion to our verification process was approximately 2 man-months. The cost associated with this effort pales in comparison to the potential disastrous monetary costs associated with replacing field failures, mask costs to re-spin the chip, and revenue lost by being late to market . UPF-based low power verification does involve extra up-front time and effort leading to a potential longer design time, but the time can be folded in under other back-end work, for no effective delay.

The integration between Mentor's front-end low power RTL simulation capability and the backend tool flow could be somewhat simpler. In particular, if the backend tools could read UPF files, this would eliminate the need for CPF to UPF conversion.

UPF also holds promise for next generation designs, which may involve techniques such as dynamic voltage scaling. UPF already has features and capabilities that would support such designs.

One currently missing capability is low power verification using emulation in order to address full system software verification. Emulation could speed up hardware-software co-verification significantly and therefore enable verification scenarios such as Linux boot and actual application runs. However, this seems likely to become available in the near future.

4. ACKNOWLEDGMENTS

The authors would like to thank Applied Micro Circuits and Mentor Graphics Corporation for supporting the development of this paper.

5. REFERENCES

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

[2] Keating, M., Flynn, D., et al. Low Power Methodology Manual for System on Chip Design. Chapter 2, Standard Low Power Methods. Springer, 2007.

[3] APM. AppliedMicro PacketPro(TM) Multicore Processor Family Provides Intelligent SoC
http://investor.appliedmicro.com/phoenix.zhtml?c=78121&p=irol-newsArticle&ID=1474906&highlight=
Sept 27, 2010.