Overcoming barriers in Power Aware Simulation

Mohit Jain  Amit Srivastava
Amit Singh  Bharti Jain
J.S.S.S.Bharath

ST Microelectronics  Mentor Graphics
Agenda

• Introduction
• Typical PA Simulation using UPF
• Barriers in PA Simulation
• Traditional approaches and their limitations
• Power Aware Models
  – Power Management Interface
  – Power Management Behavior
• Examples
  – Analog Macros
  – Macros with internal Switching
  – Voltage Regulators
• Conclusion
**Introduction**

**Today's SoCs are**
- Incredibly Complex
- Sophisticated power management for highly power efficient design
- Integrate variety of hard macros
- On-chip power sources
- Various analog IPs

**They Must**
- Verify the power management — early in the design flow
Power Aware Design and Verification

- Different Systems have different power management
- Power Gating
  - Isolation
  - Retention
- Multi-Voltage
  - Level Shifting
- Body Bias
  - Forward Bias
  - Reverse Bias
- Dynamic Voltage & Frequency Scaling
- UPF provides commands which can express the power management
IEEE 1801: Unified Power Format

- RTL is augmented with a UPF specification
  - To define the power architecture for a given implementation

- RTL + UPF drives implementation tools
  - Synthesis, place & route, etc.

- RTL + UPF also drives power-aware verification
  - Ensures that verification matches implementation
Barriers in PA Simulation

- Incomplete verification models for hard macros
  - Lack of power management information
- Variety of IPs with different power management
- Burden in switching from non-PA to PA modes
- Trade-off between accuracy and speed
Traditional Approaches & their Limitations

- **Simulation models without power management behavior**
  - incomplete design behavior of hard IPs
  - loss of simulation accuracy

- **Simulation models with power management behavior**
  - the extra supplies pose problems in integration process at RTL as the supplies are not present in regular non-pa simulations, deferring the use till post-layout netlist phase
  - inability of HDL to capture crucial power management information like power states, voltage information results in loss of significant details required for effective power aware verification

- **Analog Models**
  - Most accurate representation of the hard IP
  - Generally written in Spice and hence depend on Spice simulations
  - Extremely slow and hence limit the verification capability
Power Aware Models

- Enables Power Aware Simulation for Hard Macros
- Components of PA Model
  - Power Management interface
  - Power Management Behavior
- Power Management Interface
  - Liberty attributes
  - UPF commands
- Power Management Behavior
  - Power aware HDL
  - Derived from explicit power pin model
Power Management interface

- Provides information about power management at IP interface
- Typically captured in UPF or Liberty
- Used during integration of IP
- PM interface consists of
  - Supply information
  - Related supplies
  - Power states
  - Interface protection cells
Create Power Model Boundary
   begin_power_model hardMacro
Create Top level power domain and define supplies
   create_power_domain pd_hardIP \ 
      -include_scope \ 
      -supply { backup_ssh } \ 
      -supply { primary }
Define related supplies
   set_port_attributes \ 
      -driver_supply_set \ 
      -receiver_supply_set
Define power states
   add_power_state PD_HardIP \ 
      -state ON { \ 
         -logic_expr { \ 
            PD_HardIP.primary == ON \ 
         } \ 
      } \ 
Interface protection cells
   set_isolation PD_HardIP \ 
      -applies_to outputs
Complete the Power Model Boundary
   end_power_model
Power Management Interface
Liberty

Supply pins defined as pg_pin in liberty model

```plaintext
gp_pin(VDD) {
    pg_type : primary_power;
    voltage_name : PWR;
}
gp_pin(VSS) {
    pg_type : primary_ground;
    voltage_name : GND;
}
```

Define related supplies

```plaintext
pin(IN) {
    direction : input;
    related_power_pin : VDD;
    related_ground_pin : VSS;
    ...
}
```

Interface protection cells

```plaintext
pin (SP) {
    direction : input;
    is_isolated : true;
    isolation_enable_condition : "en";
    related_power_pin : VDD;
    related_ground_pin : VSS;
    ......
}
```
Power Management behavior

- The power management behavior is captured in HDL descriptions
- **Non-PA Behavioral Model**
  - The non-pa behavioral model doesn’t have any sensitivity to power changes
  - The power behavior is taken from related supply information to corrupt the boundary pins, which is approximate
  - Can only reflect basic power management capabilities without any retention capability. (cannot be used for IP’s with complex power behavior)
- **Allpins Model**
  - have the power pins at the port level.
  - The effect of the supplies on the outputs programmed.
  - These are traditional power aware HDL models that are used at Gate level or a later stage when netlist contains full connection of supplies.
Power Management behavior

• **PA Behavioral Model**
  - The power behavior is visible when the UPF connections are made.
  - The PA behavioral model has the following characteristics
    - The power pins are not present in the portlist.
    - The input power supplies are declared as registers and are initialized to their definitive logical values.
    - If the IP has any output supplies, they are declared as registers/wires and contain the functional intent in them.
  - Generally a PA behavioral model is derived from an allpins model by removing the supplies from the port level and declaring them as reg/wire.
Simulation of Power Aware Models

• Integration involves
  – Design integration
    • Instantiation of Behavioral Model (similar to non-PA)
  – UPF integration
    • Connection of UPF supply nets to pg_pins present on Liberty Model

• Tools can automatically combine the two components to generate PA Model
  – Power Management interface
  – Power Management behavior

• The combined version of PA model is used during simulation to provide accurate behavior

• The Power Management Behavior component can be easily reused in non-PA simulation
Examples
Analog Macros

- Mixed signal IP’s have multi-power rails
- The power behavior can be taken from
  - Liberty
  - PA HDL model
Case I: Non PA HDL and Liberty

- Behavioral Model is not power aware
- PA behavior given by liberty
  - `power_down_function`

---

```verilog
module ana_mac( ... ip1, ... );
    ... //valid macro functionality.
    //Non power aware model
    ...
    endmodule

PA Information in Liberty
pg_pin(avdd) {
    pg_type : primary_power;
}
pg_pin(dvdd) {
    pg_type : primary_power;
}
pin(REFANA) {
    ... power_down_function :"!dvdd+!advv+dvss+avss";
    ...
}
```
Simulation Results: Non PA HDL and liberty

Limitation: Actual design behavior is not captured in the simulation
Case II: PA HDL and Liberty

- The Behavioral Model is power aware
- Interface is provided by the liberty

**PA Information in Liberty**

```plaintext
pg_pin(avdd) {
    pg_type : primary_power;
}
pg_pin(dvss) {
    pg_type : primary_ground;
}
```

**PA Behavioral Model**

```plaintext
module ana_mac( ... ip1, ... );
...
reg avdd, dvdd, avss, dvss;
always @(avdd or dvdd or avss or dvss)
    begin
        if (avdd == 1'b1 && dvdd == 1'b1 && avss == 1'b0 && dvss == 1'b0) begin
            // valid macro functionality...
            end
        else begin
            // Invalid supplies, outputs are corrupted or pulled L..
            end
        end
endmodule
```
Simulation Results: PA HDL and liberty

Actual design behavior is observed in this case
Macros with Internal Switching

- Some of the IP’s have a power switch embedded inside the IP boundary for powering off some of the logic.
  - For a granular control of the power supplied to these IP’s.
  - Internally switched supply could be shared outside the IP
- The switches need to be synchronous for valid operation

© Accellera Systems Initiative
PM Interface and Behavior for internally switchable macro

UPF Connections
connect_supply_net vdd_n_lv
    -ports { hm_inst/vdd }
connect_supply_net gnd_snet
    -ports { hm_inst/vss }

PA Information in Liberty
pg_pin(vdd) {
    pg_type : primary_power;
}
pg_pin(vdd1) {
    pg_type : primary_power;
}
pg_pin(vddi) {
    pg_type : internal_power;
    direction : internal;
    switch_function : ctrl;
    pg_function : vdd;
}

PA Behavioral Model
module mac_int_sw( ... sw_ctrl ... );
    ... input sw_ctrl;
    reg vdd, vdd1,
    wire w_vddi;
always @(w_vddi) vddi = w_vddi;
initial vdd = 1'b1; vdd1 = 1'b1;
assign w_vddi = sw_ctrl === 1'b1 ?
    vdd : sw_ctrl === 1'b0 ? 1'b0 : 1'bx;
... Macro Functionality ...
endmodule

pin(ctrl) {
    direction : input;
    switch_pin : true;
}
pin(outp) {
    power_down_function : "!vddi + vss";
}
Simulation Results: Synchronization Error

Figure 10: Example depicting a scenario in which the switches are not synchronized for which the tool logs a synchronization error.

- The above snapshot depicts the case in which the controls to both the switches (internal and external) are out of sync, the simulator issues a warning indicating that a UPF net is driven by multiples sources with different states. When the switches are synchronous the UPF net will be driven to a desired state.
A voltage regulator generates a regulated and monitored supply with proper voltage level and drive capability.

- Typically consists of a control block and an analog switch.
- The regulated supplies of a voltage regulator are typically driven by analog switch (with the control of the switch governed by the regulator), with their presence inside / outside the IP boundary.
- The regulated supply can be bypassed by an external source.
### PM Interface and Behavior for Voltage Regulator

#### UPF Connections (Analog-Switch Inside)
```plaintext
create_supply_net net_sw_inside -domain vreg_domain -resolve one_hot
connect_supply_net net_sw_inside -ports vreg_inst/vdd_sw_inside
```

#### UPF Connections (Analog-Switch Outside)
```plaintext
create_supply_net net_sw_outside -domain vreg_domain -resolve one_hot
connect_supply_net net_sw_outside -ports vreg_inst/vdd_sw_outside
connect_supply_net net_sw_outside -ports ana_mac_sw_inst/vdd_driver
```

#### PA Information in Liberty for VREG
```plaintext
is_macro_cell : true;
pg_pin(vdd_sw_outside){
  voltage_name : "vdd_switch_outside";
  pg_type : primary_power;
}
pg_pin(vdd_sw_inside){
  voltage_name : "vdd_switch_inside";
  pg_type : "internal_power";
  direction : inout;
}
```
Simulation Results: Voltage Regulator

- The simulation snapshot shows two cases:
  - When the regulator is in regulation mode
  - When the regulator is in Bypass mode
- There could be another scenario in which the ballast is inside the IP interface, the results will be similar.
Conclusion

• PA Models enable accurate PA-simulations
• Complex issues are caught early at RTL design phase
• PA Models are incomplete in terms of understanding voltage values
• Future homes on Voltage Aware Models
  – the more accurate means of reflecting design behavior
Thank You

Questions