Methodology for Power Implementation and Validation at Higher Level of Abstraction

Amit Kumar Awasthy
M. Tech (Digital Communication)
Maulana Azad National Institute of Technology
Bhopal (M.P.), India

Lalita Gupta, Ph.D
Assistant Professor
Maulana Azad National Institute of Technology
Bhopal (M.P.), India

ABSTRACT
In earlier generation of IC design technologies the prime parameters of concern were timing and silicon area. The increasing demand for high-performance, battery-operated, system-on-chips in communication and computing has shifted the focus from traditional constraints (such as area, performance, cost, and reliability) to power consumption [1]. In recent years device densities and clock frequency have increased dramatically in devices thereby increasing the power consumption dramatically. During design process the most critical power requirements are tested only after power pins becomes explicit. There are different design strategies for reducing power consumption, and it also becomes critical to make power aware design even if power pins were not explicit or say at very abstraction level of design flow. UPF is used as an IEEE 1801 standard method which provides a consistent way to specifying power implementation intent throughout the design process [2]. Low power validation at very abstraction level uses RTL/Behavioral HDL model along with UPF intent. UPF provide an ability to verify the power intent behavior as early as possible or say at higher abstraction level by overlay the power behavior over the RTL/Behavior HDL model at same abstraction level. This paper describes how HDL model impacted at very higher abstraction level to meet certain power constraints and their validation using an industry accepted IEEE 1801 standard UPF low power validation flow.

General Terms
Power functionality, Abstraction level, Design flow, Power consumption, Logical functionality, Power implementation

Keywords
HDL (Hardware Description Language), RTL (Register Transfer Level), UPF (Unified Power Format), EDA (Electronic Design Automation), CAD (Computer Aided Design), TCL (Tool Command Language).

1. INTRODUCTION
As design complexity increases Hardware Descriptive Language is widely used to replicate the design and operation of electronic circuit at higher level of abstraction [3]. There are two most popular and widely used HDL are Verilog HDL and VHDL. The electronic design can be implemented in different level of abstraction using HDL, Behavioral and RTL are higher level of abstraction. In Behavioral level we represent our electronic design in terms of algorithms. At RTL abstraction level design can be represented in terms of data flow between different registers.

Traditional HDL models have only logical functionality information about the design as only logical pins are present on top port list. Power functionality comes into picture only when power and ground pins become explicit during placement and routing stage. The idea of power aware HDL description enable us to optimize the power consumption at higher level of abstraction. Power functionality is just like a wrapper over the logical functionality if power and ground pin status is not true then the logical functionality has no significance. The only challenge is that at higher level of abstraction the power and ground pins are not present at top port list which makes generation and validation of power implemented HDL model quite trickier.

2. POWER CONSUMPTION
Designers must consider two types of power consumption, Dynamic and Static [1]. Dynamic power is consumed during the switching of transistors, so it depends on the clock frequency and switching activity. Static power is the transistor leakage current that flows whenever power is applied to the device, so it is not related to the clock frequency or switching activity. There are different methods for reducing the power consumption are Insertion of power down modes in design, clock gating, power gating, reducing supply voltage, multiple threshold libraries and voltage and frequency scaling.

3. TRADITIONAL HDL MODEL AT HIGHER LEVEL OF ABSTRACTION
Traditionally HDL models have only logical functionality information no power information is given through HDL models. At higher level of abstraction the design functionality can be given in terms of truth table. If **Z** is the output and **i1** and **i2** are logical inputs then the output Z and input i1 and i2 are related as:

$$Z = f(i1, i2)$$

This shows that the output is only function of logical inputs, so the design functionality is equivalent to the logical functionality of electronic design.

Table 1. Truth table of logical design functionality

logical input		output
i1	**i2**	**Z**
0	0	0
0	1	0
1	0	0
1	1	1

The truth table (Table 1) represents the algorithmic description of electronic design is given at higher level of abstraction. Only interface information and design

functionality in terms of truth table given at this particular abstraction level. The electronic design at higher level of abstraction is represented as a black box (shown in Figure 1) with interface information.

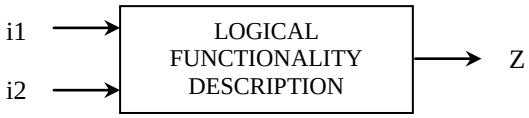

Figure 1: HDL model description as block diagram

The HDL description is consisting of module declaration and module definition [3]. The traditional HDL model has only logical functionality and logical pins.

```
module traditional_hdl_model (Z, i1, i2);
output Z;
input i1;
input i2;
.
.
Logical
Functionality
Description
.
.
endmodule
```

4. POWER IMPLEMENTATION AT HIGHR LEVEL OF ABSTRACTION

If power implementation done at higher level of abstraction then the design functionality consist both logical as well as power functionality.

If **Z** is output, **i1** and **i2** are logical input, **sup** is power supply and **gnd** is ground pin then the output is given as:

Z = f (i1, i2, sup, gnd)

This shows that the output is a function of not only logical inputs but also the power and ground pins. The design functionality consist both logical as well as power functionality. The complete functionality of the design in form of truth table is given in Table 2.

Table 2. Truth table of complete design functionality

Power supply	Ground	Logical input		Output
sup	**gnd**	**i1**	**i2**	**Z**
0	-	-	-	X
-	1	-	-	X
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1

As per Table 2 when power pin is in high logic state and ground pin is in low logic state then only logical functionality comes into picture, if power and ground pins are not connected properly then the logical functionality has no significance. The usage of power implemented HDL model with complete functionality or for low power validation can be shown by flow diagram in Figure 2.

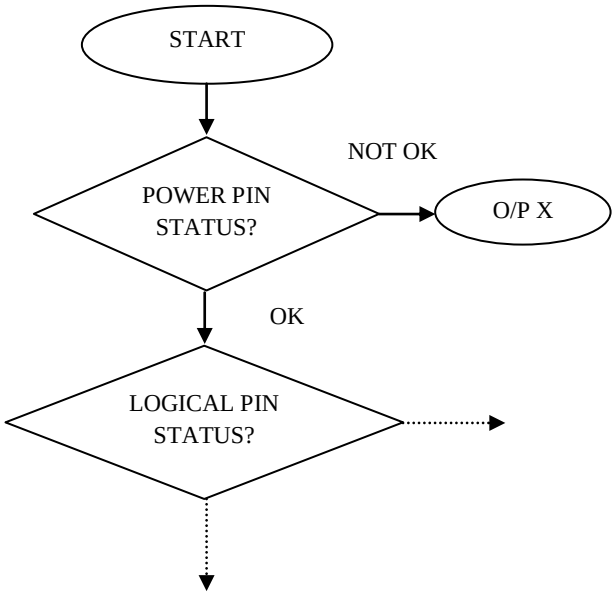

Figure 2: Flow for low power validation of power implemented HDL model

If power functionality also present with logical functionality then first we have to check power and ground pin status if everything is in true state then only logical functionality behavior of design comes into picture as shown in figure 2. At higher abstraction level power and ground pin is not in top port list so a power and ground pin logic generation block needed within the HDL description itself so that the model becomes power implemented model which also used for functional validation just like traditional non power aware HDL model. This power and ground pin logic generation block helps to skip the checking of power pins logic as shown in figure 3.

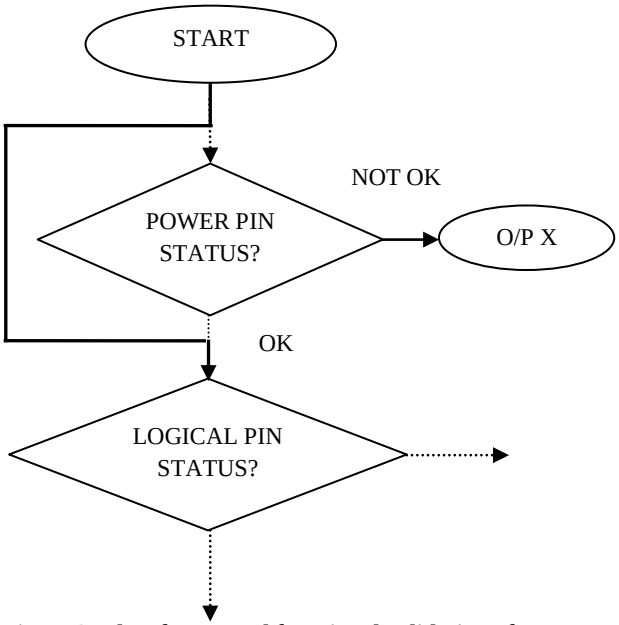

Figure 3: Flow for normal functional validation of power implemented HDL model

The HDL model after implementation of power functionality at higher level of abstraction consist the complete functionality of design and a power & ground pin logic generation block. The power implemented HDL model is given as:

```
module power_imple_hdl_model (Z, i1, i2);
output Z;
input i1;
input i2;
.
.
Internal
Supply & ground
Generation
.
.
.
.
Logical + power
Functionality
Description
.
.
endmodule
```

5. VALIDATION OF POWER FUNCTIONALITY AT HIGHER LEVEL OF ABSTRACTION USING UPF

Generation of power implemented model is quite trickier task but validating the power functionality at this abstraction level is the toughest task as direct port to port contact for power and ground pins are not possible as they are not present at top port list. Those power and ground pins are defined as internal variable in the module description. UPF (Unified Power Format) provides a way to validate the power implemented model by creating the power switches, power nets and connecting those nets directly to the domain supply of the design [2]. In UPF low power validation flow a combination of HDL and UPF intent are used, the UPF intent is nothing but a collection of TCL (Tool Command Language) commands. UPF is an IEEE 1801 standard for validating the low power integrated circuits. The UPF validation process includes:

- Power switch creation
- Power domain creation
- Supply net creation
- Supply port creation
- Power controller creation
- Connecting all nets to respective ports

All these things at CAD level can be created and connected by using TCL commands. After creating all these things a power controller needed to validate the design in terms of power functionality by toggling the power and ground pin. Power controller block can be invoked through some variable used in test bench. For UPF validation there are different EDA tools available in market like Incisive Enterprise Simulator (IES) from cadence, Native VCS from Synopsys.

6. RESULTS

The validation result of logical functionality for traditional HDL model shown in figure 4. The waveform in figure 4 has two logical input i1, i2 and one output Z. This type of validation is known as functional validation.

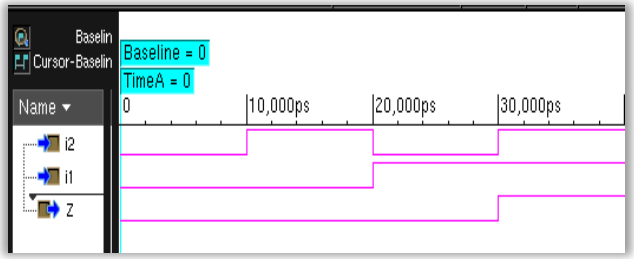

Figure 4: Functional validation result of traditional HDL model

The internal schematic generated through HDL model (given in figure 5) consist only one internal algorithmic block as this type of model is used for higher level of abstraction any hardware information is not given by internal schematic view.

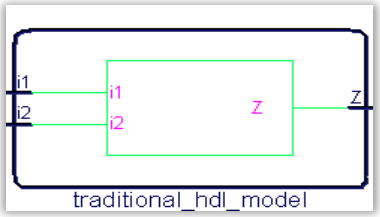

Figure 5: Internal schematic view of traditional non power implemented HDL model

After implementation of power at very abstraction level need of UPF validation comes into picture along with functional validation. The waveform of the validation of power implemented model is given in figure 6. In waveform we have two inputs i1, i2 one output and some internally generated variable which is generated through power generator block. Till 40,000 ps the validation is similar to functional validation as power supply and reference ground is connected properly. At 40,000 ps the power supply goes down, as power supply goes down the output is no longer depends on input. After 40,000 ps output becomes X (don't care) and any change in input is not affecting the output, Hence reducing the power dissipation.

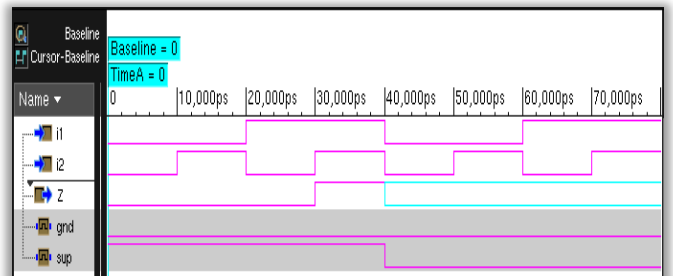

Figure 6: UPF validation result of power implemented HDL model

The internal schematic can be given in figure 7, the circled block is known as power generator block. The power generator block is totally internal to the schematic and has not connected from outside. UPF intent has a capability to control the power generator block through power controller block which is written through UPF intent.

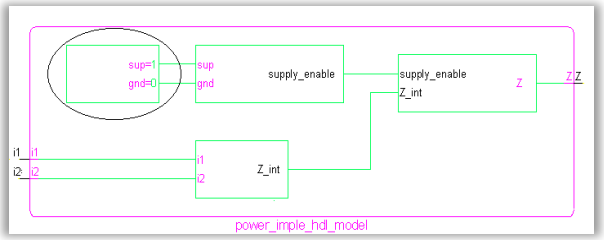

Figure 7: Internal schematic view of power implemented HDL model

With addition of power functionality the HDL model and its internal schematic becomes quite complex, but this method enable us to validate power functionality at higher abstraction level of design flow.

7. CONCLUSIONS

On conclusion UPF is a good way to validate the power functionality of power implemented HDL model at higher abstraction level of design flow which reduces great efforts in further design process and saves money as well as time. The power implemented model can be used for both validation purpose either directed functional validation or for UPF low power validation.

8. REFERENCES

[1] Version D-2010.03, March 2010. Synopsys Low-Power Flow User Guide

[2] IEEE Std. 1801™-2009 reference manual, March 2009. IEEE Standard for Design and Verification of Low Power Integrated Circuits

[3] Samir palnitkar, second edition. Verilog HDL: A Guide to Digital Design and Synthesis by Prentice Hall PTR

[4] Kanika Kaur et. al. "Strategies & methodologies for low power vlsi designs: a review" IJAET may 2011 Vol. 1,Issue 2,pp.159-165

[5] Enrico Macii et. al. "Power modeling, estimation, and optimization", IEEE transactions on computer-aided design of integrated circuits and systems, vol. 17, no. 11, november 1998 [6] J. M. Rabaey and M. Pedram, Eds., "Low Power Design Methodologies". Norwell, MA: Kluwer Academic, 1996

[7] D. Singh et. al. "Power conscious CAD tools and methodologies: A perspective" Proc. of the IEEE, 83(4): 570-593, 1995.

[8] Mark Horowitz et.al. "Low-Power Digital Design" Center for Integrated Systems, Stanford University, Stanford, CA 94305