Power Calculation and Modeling in Deep Submicron

Jay Abraham
Silicon Integration Initiative (Si2)
4030 West Braker Lane, Suite 550
Austin, TX 78759, USA
1 (512) 342-2244
jabraham@si2.org

1. ABSTRACT

Over the past few years it has become increasingly apparent that modern IC design is no longer bounded by timing and area constraints. Power has become significantly more important. In an era of hand held devices ranging from mobile computing to wireless communication systems, managing and controlling power takes on an important role. Several benefits are realized with low power designs in addition to extended battery life. Low power devices often run at lower junction temperatures and this leads to high reliability and low cost cooling systems [1,2,3,6]. Calculation and modeling of power (and delay) in deep-submicron (less then 0.25 microns) designs poses several challenges. This paper discusses the use of the Delay and Power Calculation System (DPCS) as a means by which EDA (Electronic Design Automation) tools can accurately calculate and model power.

2. INTRODUCTION

DPCS standardized by OVI (Open Verilog International) and Si2 (Silicon Integration Initiative) attempts to address the problems of power and delay calculation with the use of a common library system [4]. The DPCS standard is currently under IEEE review for formal standardization. DPCS has been assigned Project Authorization Request number 1481 [5].

ISLPED98, Monterey, CA, USA

The motivation for DPCS is that today there is no consistent standard for the modeling and/or calculation of power and delay. Formats in existence are either inadequate or propriety. With the DPCS, the Silicon Provider creates a single library that contains very accurate power and timing modeling information (labeled as the Common Delay & Power Calculation Expression Language as shown in fig.1). This text file is compiled with the DPCS compiler to produce a compiled library known as the DPCM (Delay and Power Calculation Module). The DPCM is a binary executable module that contains power and delay modeling and calculation capabilities including the ability to compute detailed interconnect delays. All EDA tools dynamically load the DPCM. Power and delay information is extracted from the DPCM via an Application Procedural Interface (API). The use of such a system ensures consistent power and timing analysis. In summary, the DPC system consists of an expression language (referred to as the DCL or Delay Calculation Language) and an Application Procedural Interface.

3. DELAY AND POWER CALCULATION

With DPCS the accuracy of describing timing and power is left up to the IC vendor. Traditionally the IC vendor is the one that has always known how to best describe these characteristics. The advent of commercial EDA tools meant that the EDA vendor controlled delay and power modeling. The result was an endless loop in which the IC vendor would have to run his/her sign off tools against a design to

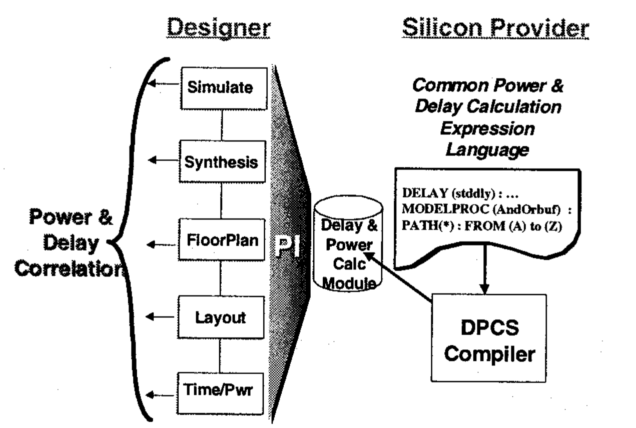

Figure 1: DPCS power and timing methodology

ensure that it had adequate timing and power margins. DPCS eliminates this, because any EDA tool is able to access the IC vendor's propriety delay and power calculation algorithms that are embedded within the DPCM. The IC vendor has the flexibility to implement any type of power calculation and modeling algorithm (e.g. table lookup with interpolation or propriety equations). Since the DPCM is compiled, IP protection of algorithms is assured.

3.1 Interdependence of Delay and Power

Accurate power calculation can heavily depend on delay calculation, specifically with regards to ramp or slew propagation. Output slews of cells are often dependent on input slews and output driving capacitance. As a result of this it is often necessary to propagate slews from a primary input to a primary output for the circuit under evaluation. Wire slew can in turn be dependent on wire delay as well. Power consumption for a typical cell is often a function of input slews, output capacitance, transition levels or conditions on input pins and other vendor specific process parameters. As a result of this, in order to compute accurate power consumption, it may be necessary to compute slews as well. DPCS takes advantage of this interdependence in that it provides calculation for slews and power in a single medium or form.

3.1.1 Slew Propagation

Initially before an application (in this case application refers to any EDA application, e.g. a simulator, power calculator etc.) requests for cell power consumption, the application will request the DPCM to model a cell or block. During the model phase, the DPCM educates the application of all possible timing arcs and segments that are available. These timing segments will be used for slew propagation. In addition, the DPCM also describes propagation edges for given timing segments (e.g. rising waveform on an input signal resulting in a falling waveform on the output signal etc.). All of this information is encoded within a DPCM and is accessed via the application as shown in fig. 2.

After modeling a cell the application requests the DPCM to

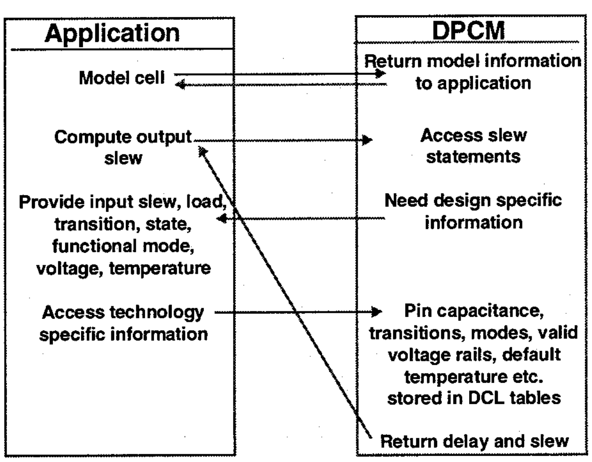

Figure 2: Slew calculation for slew propagation

compute output slew for a given timing segment or arc. The DPCM then invokes the IC vendors propriety slew calculation algorithm. This algorithm may require the knowledge of design specific information such as input slews, current cell state, input pin condition or transition information etc. The application can rely on the DPCM for technology specific information such as pin capacitance, default wire load tables etc. for computation of capacitive load, default voltage and temperature and so forth. Once these parameters are obtained, the application communicates this information to the DPCM via the standards defined API. The DPCM then uses this information to compute output slew and returns control to the calling application.

4. POWER MODELING

Unlike timing modeling, power modeling for the first revision of the IEEE standard consists of 'black box modeling'. For timing, all timing segments within a cell (this can be thought of as a block or black box) are enumerated. For power modeling, the application is unaware of the contents of a cell. It does not need to know about timing arcs etc. Instead the application is informed of any data associated with the specific cell under investigation. Power consumption is now solely based on input slews, output capacitance and pin transitions. The library developer creating the DPCM can have separate modeling statements for slew (inherently this is the model for timing or delay) and power. Appropriate language constructs will inform the application via the API of the library developer's choice.

4.1 Power Calculation

Three modes of power calculation are supported in DPCS. These are termed AET or All Events Trace, Group, and Pin power. These vary from computational accuracy and execution speed depending on the type of information that is available. It is the DPCM library developer's choice to implement one, two or all three of the calculation methods. During the initial handshaking process between the application and DPCM, the application will determine if the DPCM supports the required power calculation method.

During the modeling phase, the DPCM will provide to the application *handles* by which the application can use to access data specific to a cell or block. This data consists of valid transitions and pins that application is to monitor, and initial state choices. This type of data is accessed after modeling is completed as shown in fig. 3.

The power computed by the DPCM is in terms *of static power* and/or *dynamic energy*. Static power is defined as power consumed by the cell for the state that it has just transitioned ***into***. Depending on the technology, static power can be classified as leakage power as well as any analog power due to pull ups and pull downs. Dynamic energy is defined as the energy expended during the transition of one state to another. This consists of crowbar power and capacitive load power [3]. Dynamic power is computed by the application using the following equation:

(dynamic energy captured during the transition) - (static power for the state transitioning INTO) * (time of transition)

The dynamic energy returned by the DPCM does ***not*** include static leakage power. The application computes total power consumption, time-averaged, maximum, RMS, and peak power by accumulating power over time. The three power modeling and calculation features of DPCS will now be explained:

1) In AET (All Events Trace) mode the application passes all changes or conditions on the input pins of the cell or block under examination to the DPCM. The application is first given a list of pins which need to be monitored. Conditions are evaluated whenever there is a voltage swing on any of the identified pins (e.g. 0 to 1, 1 to 0, 0 to HiZ, 1 to X etc.). With this method, the DPCM may keep its own representation of the previous states (i.e. state history). Power is then computed for the specific pin change and data is returned to the calling application (see fig. 3).

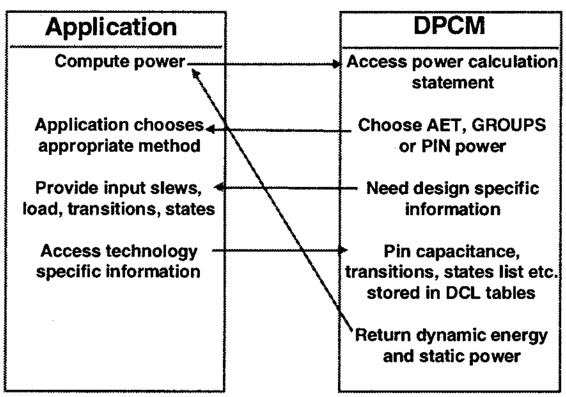

Figure 3: Modeling and calculating power

2) In Group power mode the application is given a list (or group) of pins and valid states and conditions that the application must monitor. When specific conditions are met (become true), the application calls for power for the power consumed by that condition. Alternatively, the application can accumulate which conditions are true and call for power calculations at some later point, multiplying the returned power by the number of times each condition was true. With this method, the DPCM informs the application of the pins and the associated conditions (or states) of those pins which are important for power computation. The application is then responsible to determine which conditions are true, and to call the DPCM for power on each one. Chronological ordering of these calls is not required and the DPCM cannot use the chronological ordering of power calculation requests to keep its own representation of previous states. The power returned is the power of the cell specific to the particular condition chosen.

3) In Pin power mode the application calls for power for each pin on the cell that has had a change in condition or state. Power is calculated without any information of the cell's previous or present state. The DPCM is only passed the knowledge that a pin changed. Chronological ordering of these power calls is not required. The DPCM cannot use the chronological ordering of power calculation requests to keep its own representation of previous states. The power returned is the power of the cell for the specified pin change.

DPCS also defines two terms, "Simultaneous Switching Time Window" and "Settling Time Window". Simultaneous Switching Time Window is the time interval when two events are treated as one event. If the time between two events is less than or equal to the simultaneous switching time window, the application calls for power on only one event. It is the application's choice as to which event it chooses. Settling Time Window is defined as the time interval when two events are treated as a partial swing. If the time between two events is less than or equal to the settling time window, then the application calls for a special API for the calculation of partial swing energy for this event instead of group power.

5. CONCLUSION

DPCS is a mature and proven technology that has been in use by IBM Corporation for several years. DPCS compilers, products and support are available today. DPCS has proven to be a viable form of representing timing and power for future generation of ICs. It is expected that several DPCS compliant applications will be available from major EDA vendors very soon as well as the availability of DPCS supported design kits from various silicon suppliers.

6. REFERENCES

[1] Jerry Frenkil, "Low-Power ASIC Design," *EDN*, March 15,1997.

[2] Enrico Macii, Massoud Pedram, Fabio Somenzi, "High-Level Power Modeling, Estimation, and optimization," DAC 97, June 1997.

[3] Jerry Frenkil, "Tools and Methodologies for Low Power Design", DAC 97, June 1997.

[4] OVI and Si2, "Delay Calculation Language and Procedural Interface Specifications," *Silicon Integration Initiative*, Version 1.0.2, September 1997.

[5] IEEE 1481 draft specification "Delay and Power Calculation System," unpublished.

[6] Jianwen Zhu, Poonam Agrawal, Daniel Gajski, "RT Level Power Analysis", ASP-DAC 97, January 1997.