

A Methodology to Analyze Power, Voltage Drop and Their Effects on Clock Skew/Delay in Early Stages of Design

Masato Iwabuchi, Noboru Sakamoto, Yasushi Sekine and Takashi Omachi

Hitachi, Ltd., 6-16-3 Shinmachi, Ome, Tokyo 198-8512, Japan

Abstract

This paper presents a methodology to analyze signal integrity such as power voltage drop and clock skew in early stages of design, more specifically, when RTL-design and early floorplanning are performed. In this stage, logic contents are not known, but global structure of power/ground and clock networks, function partitioning and early floorplan give reasonable accuracy for global optimization of the chip. A case study shows the power voltage drop and critical path delay slowdown due to dynamic power voltage drop for a mixed analog-digital chip, and a good match with actual measurements is achieved.

1. Introduction

Several design issues need to be addressed while moving into Deep Submicron (DSM) design [1].

- 1) Capacity (number of transistors in a chip) is increasing.
- 2) Speed (clock frequency) is increasing.
- 3) Wiring pitch is decreasing and hence interconnect delay is increasing and becomes dominant.
- 4) Demand for reducing power is increasing.
- 5) Pressure of reducing time-to-market is increasing.

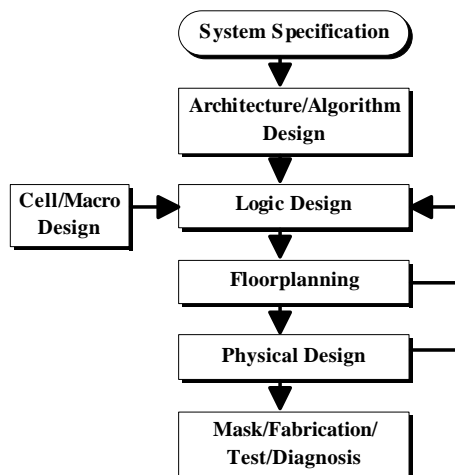


Fig. 1 A typical conventional DSM design flow

Moving into DSM design, signal integrity such as power voltage drop and clock skew has a great impact on other design targets - timing, power dissipation, and area. Higher device density and

faster switching speed cause larger current in the power network, thus causing performance degradation due to the voltage drop in the power network. As clock frequency increases, the robustness of the power network and clock tree becomes increasingly required. However, robustness becomes a trade-off against competing design targets, such as area.

A typical design flow used for DSM design is shown in Fig. 1. However, the flow shown in Fig. 1 does not highlight signal integrity analysis. Often parasitic extraction and signal integrity verification is done after the physical design, but this has several problems.

One problem is that in designing the power network and clock tree, the decisions about the power network and clock tree structure have to be made at the very early stages before having a gate-level netlist. Most commercial tools and previous works focus on when the entire chip design is complete and detailed information about the design are known. Thus one key technology is how to understand and capture information about the design at the very early design cycle, and to model it with reasonable accuracy for use in analysis.

A second problem is that if we only analyze signal integrity when the entire chip design is complete and detailed information about the design are known, then the design cycle time is too long. If design problems are found at the last design stage, they are usually very difficult or expensive to fix. It is desirable to find problems as early as possible, especially at the early design stage, where design change is less expensive.

Exploring a large number of trade-offs between robustness of power/clock distribution and area is also a key in achieving high design quality (small area, high speed, low power dissipation). This is another reason why the turn around cycle time for the signal integrity analysis must be very short.

A third problem is computational time for calculating power voltage drop. Since power voltage drop is a global chip problem, it cannot be solved independently in partitioned parts of the chip. However, for a large design in DSM, it is not computationally feasible to simulate a whole chip at the device level. This is another reason that early evaluation is more important rather than extracting devices and interconnect RCs from the whole chip and simulating the extracted netlist.

This paper presents a methodology to design and analyze power network and clock tree in early design stage, more specifically when RTL-design and early floorplanning are performed.

This paper is organized as follows. Section 2 gives some

preliminary and a problem definition. Section 3 discusses a detailed design flow that analyzes power, power voltage drop and their effects on clock skew/delay. Section 4 discusses the accuracy and advantages of this flow. A case study of this flow is shown in Section 5, and the paper concludes with a discussion of the open issues.

2. Problem definition

A proposed design flow is shown in Fig. 2.

In this paper, we define the shaded area of Fig. 2 as "an early design stage", where,

- System specification and design targets (clock frequency, power dissipation, and area) are given,
 - Function partitioning is roughly done,
- and
- RTL-design,
 - Early floorplanning (placement of function blocks, global interconnects between those blocks),
 - Preliminary circuit design for critical path circuits such as data path and memories,
 - Clock/Power planning (Physical architecture design),
- are performed. Thus in this design stage, gate-level netlist is not yet available.

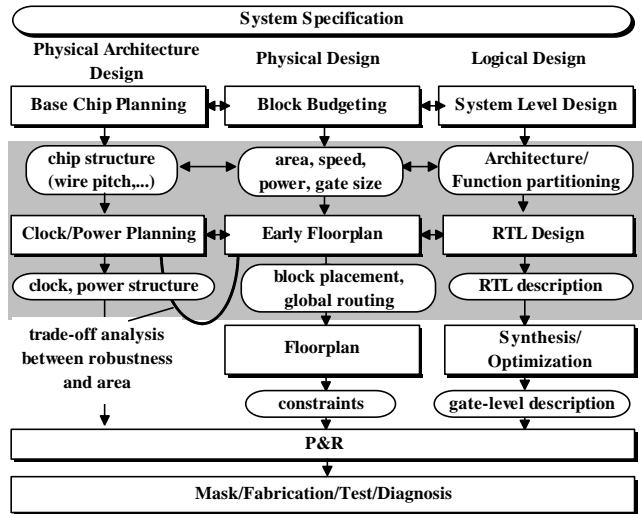


Fig. 2 A proposed design flow

From the viewpoint of signal integrity, we classify signal integrity analysis into three design phases [6].

Level 1: An early design stage, i.e. physical architecture design and early (pre-synthesis) floorplan phase.

Level 2: Post-floorplan, or in place & route (P&R) phase.

Level 3: Post-layout phase.

Our objectives in this paper is to analyze power voltage drop and clock skew in Level1 and to determine the physical architecture such as

- Global power structure (grid Vs fish bone, width and space, number of metals used, location of PADs...)
- Global clock structure (H-tree / balanced tree / grid etc, width and space, number of metals used, shielding, gated,...)

In power network analysis, not only static power drop analysis but

also dynamic analysis is necessary because of inductance effects etc. Also, since the clock is a major power consuming component, and a power voltage drop significantly effects clock skew, clock and power network planning must be tightly integrated.

We will explain the design flow in the next section. It should be noted that we will mainly explain the flow by the manner that logical design determines physical design, however, the logical design and physical design is very interactive, so that in some cases the budget determined by the physical design (a block area, for example) determines a target to the logical design (gate-counts for the block).

3. Design Flow

Step1. Block budgeting

Budgets (area, number of gates, and number of flip flops (F/Fs)) for each function block are estimated, as shown in Table 1.

Table 1. Budgets for blocks

Block		Area [mm ²]	Number of gates	Number of F/Fs
Function	Name			
	Block ₁	A ₁	Ng ₁	Nf ₁
	Block ₂	A ₂	Ng ₂	Nf ₂
		

where A_i is the area of Block_{*i*}, Ng_i is the number of gates in Block_{*i*}, Nf_i is the number of F/Fs in Block_{*i*}.

When multiple clocks are used, Nf_i is partitioned into Nf_{i1}, Nf_{i2}, \dots , where Nf_{ij} is the number of F/Fs corresponding to clock CLK_{*j*}.

The area may be estimated from the number of gates as,

$$A_i = Ng_i * \gamma \quad (1)$$

where γ is an empirical ratio between area and number of gates.

The number of gates may be estimated from previous generation designs or similar designs, or may be estimated by using Quick Synthesis. For regular structures such as memory and data path core, more precise estimations may be done, by considering their function. For memories, block area is estimated based on bits/words size, memory architecture, such as functions (1 port, multi-port), and critical path circuits. For data path cores, the critical path circuit in consideration is used to estimate block area, gate counts, and the number of F/Fs.

Note that area may not just be estimated from gate-counts. Sometimes the area for blocks are constrained by the total chip area and floorplanning requirement, and Equation (1) is used to budget the target gate-counts for gate-level logical implementation.

Step2. Floorplanning

Block are placed, considering global interconnects between those blocks, multiple clocks that are fed to the blocks.

Step3. Clock planning

A large number of previous works regarding the implementation of clock networks have been reported[2][3].In this paper, we will just describe an example clock implementation, which was used in

the case study in section 5.

We will assume our clock tree has global/local stages. The global/local clock structure is illustrated in Fig. 3. The global clock tree feeds clocks to the macro/block level, and the local clock tree feeds clocks to buffers and flip flops (F/Fs) inside blocks.

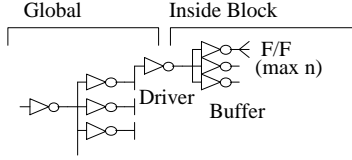


Fig. 3. Global/Local clock structure

The clock structure to be verified in our analysis is constructed as below.

1° Estimate the number of Buffers that drive F/Fs in each block.

The number of F/Fs in Block_i, Nb_i, is given by

$$Nb_i = Nf_i / \beta \quad (2)$$

where β is the number of F/Fs that one Buffer can drive.

2° Select a global clock tree structure to be analyzed.

2-1° Determine the number of Drivers that drive Buffers.

The number of Drivers that drive Buffers in Block_i, Nd_i, is given by

$$Nd_i = (C_{dwire_load} + C_{buffer}) * Nb_i / CL_{d\ max} \quad (3)$$

where C_{dwire_load} is the wire load capacitance which is driven by the Drivers, C_{buffer} is input pin load capacitance of a Buffer, $CL_{d\ max}$ is the maximum load capacitance that one Driver can drive. C_{dwire_load} is given by

$$C_{dwire_load} = L_{drive_buffer} * \hat{\lambda} \quad (4)$$

where $\hat{\lambda}$ is a unit capacitance of wires, and L_{drive_buffer} is the wire length from a Driver to a Buffer.

L_{drive_buffer} is estimated by

$$L_{drive_buffer} = \sqrt{A_i / Nd_i'} - \sqrt{Ab_i} \quad (5)$$

where $Nd_i' = (C_{buffer}) * Nb_i / CL_{d\ max}$, and $Ab_i = A_i / Nb_i$ is the area for one Buffer in Block_i.

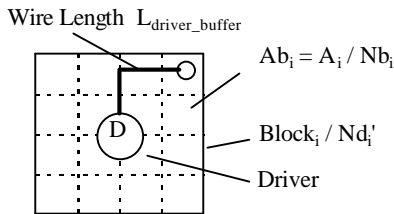


Fig. 4. Wire length estimation from Driver to Buffer

If $Nd_i < 1$, then some adjacent blocks are merged and one Driver drives them.

2-2° Select a global clock tree architecture from Clock source to Driver.

Global clock tree structures to be considered in this phase are, topology (H-tree/ balanced tree / grid etc), width and space, number of metals used, etc.

2-3° Insert gated clock

When gated clocking is used, insert a control signal to the clock tree at Driver-level so that it turns on/off a Driver. If more refined control is required, insert a control signal at Buffer-level.

3° Clock tree analysis

Analyze clock timing, clock skew without power supply voltage drop. Note that cross coupling effect between clocks and other signals is another big signal integrity issue. This is beyond the scope of this paper, however, and we just use clock shielding as much as possible.

Step4. Power estimation

Based on the clock tree in Step 3, power dissipation is estimated as

$$Power = P_{global_clock} + \sum P_i \quad (6)$$

where P_{global_clock} is the power consumed in the clock tree from

Clock source to Driver-level, and P_i is the power dissipation of Block_i.

P_i may be estimated as follows.

1) For logic blocks,

$$P_i = P_{clock}(i) + P_{toggle}(i) \quad (7)$$

$$P_{clock}(i) = Nb_i * k1 + Nf_i * k2 \quad (8)$$

$$P_{toggle}(i) = Ng_i * k3 * \alpha_i \quad (9)$$

where $P_{clock}(i)$ represents the power consumed in clock tree and their leaves in Block_i, $P_{toggle}(i)$ represents the power consumed in logic switching in Block_i, $k1$ is the power dissipation for a Buffer with average load, $k2$ is the power dissipation inside a F/F at every clock cycle regardless of output signal switching, $k3$ is the average power dissipation of gates in Block_i with average load, and α_i is the toggle rate (switching factor) for gates in Block_i.

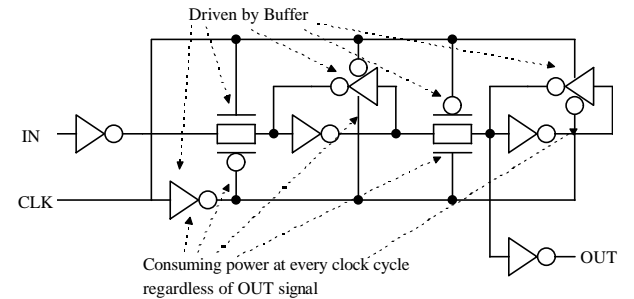


Fig.5. Power dissipation in a F/F

2)RAM core, I/O block are estimated from previous generation designs or similar designs, or may be estimated by considering memory bits/words size, memory architecture, and critical path circuits.

Step 5. Power network analysis

Based on the information of above steps, we can analyze a power network. In analyzing a power network, not only static (DC) power analysis but also dynamic (AC) power analysis is necessary for the reasons given below:

- 1) Inductance effect for power voltage drop.
- 2) On chip decoupling effect for power voltage drop.
- 3) Power voltage noise via substrate.
- 4) When gated clock is used and if a large block turns on/off, a large switching current occurs. Inductive noise caused by this switching has to be analyzed.
- 5) Power voltage drop effects clock skew and delay significantly. Later on in the design, static analysis may be used for checks of all paths for quick turn around time. But the correlation between static and dynamic analysis must be verified in this stage.

In order to analyze power network with reasonable accuracy and short turn around time, two techniques, modeling and coarse grid approach, are the key.

Analyzing dynamic power voltage fluctuation for the entire large chip at the device level is not computationally feasible. However, since power voltage drop is a global chip problem, it cannot be solved independently in partitioned parts of the chip. Thus modeling techniques that partitions a chip into coarse grids and models the power/ground networks and clock tree, pre-synthesized logic contents, are necessary. This modeling is carefully done taking into account the capacity limitation of the dynamic transistor-level simulator.

Global power/ground networks and clock tree are inter-blocks level and are usually assigned higher metal layers. Local power/ground networks and clock tree are below the block level, and are usually assigned lower metal layers. Fig. 6 illustrates information about power/ground, clock tree structures and block placement which is well captured by using a coarse grid.

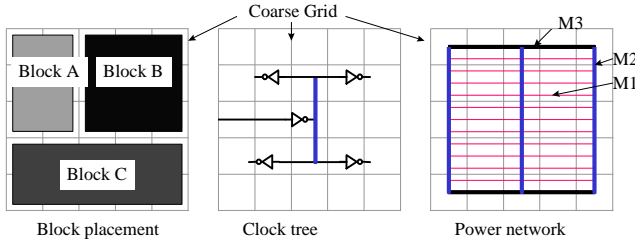


Fig. 6 Coarse grid in a chip

1. Static power analysis

1-1° Extract R from power/ground network.

We use a power structure shown in Fig. 7(a) as an example for explanation. M3, M2 is extracted in each coarse grid, and local M1 power net is extracted and reduced to a R component model per each coarse grid as shown in Fig. 7(b).

1-2° For each coarse grid, put a current source that corresponds to the global clock. This part corresponds to the power $P_{\text{global_clock}}$.

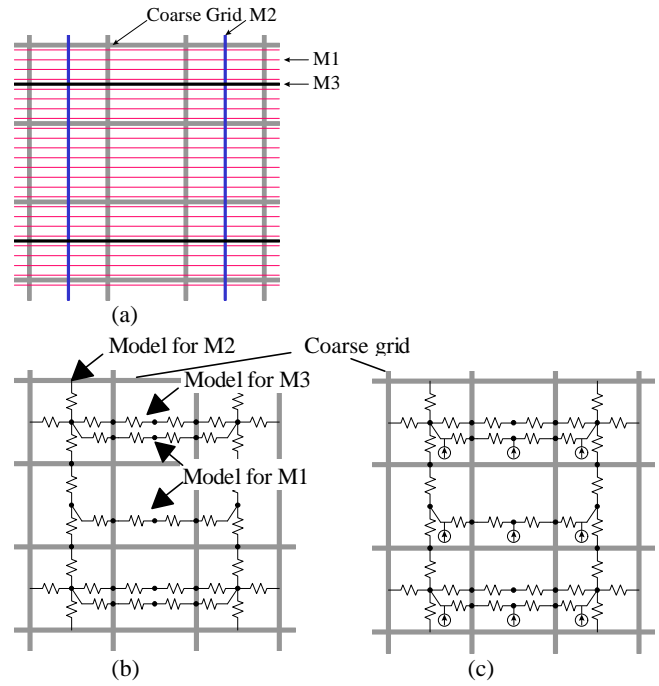


Fig.7 Network for DC power analysis

1-3° Calculate the average current $I_{\text{ave},i}$ of Block_i from P_i by

$$I_{\text{ave},i} = P_i / V_{\text{dd}} \quad (10)$$

If we assume homogeneous current distribution inside Block_i, then the current in each grid is given by

$$I_{\text{ave},i_grid} = I_{\text{ave},i} / (A_i / A_{\text{grid}}) \quad (11)$$

where A_i is the area of Block_i, and A_{grid} is the area of one coarse grid. Assign I_{ave,i_grid} to each coarse grid and put them on the power network as current sources. (Fig.7(c))

Instead of $I_{\text{ave},i}$, peak current $I_{\text{peak},i}$ may be used,

$$\text{where } I_{\text{peak},i} = \text{const} * I_{\text{ave},i}.$$

1-4° Perform static power analysis.

Electromigration and static power IR drop can be analyzed.

2. Dynamic power analysis

2-1° Extract RC from the Power/Ground network. (Fig. 8(a))

2-2° Create models for the following important components for the power netlist and attach them to the power/ground netlist.

- 1) PAD, package LRC
- 2) Substrate RC model (either mesh, or very simple R is used.)
- 3) Inherent capacitance of well-junction and non-switching devices. The inherent capacitance C_{dc} is estimated by $C_{\text{dc}} = \text{area of grid} * \text{empirical ratio}$.
- 4) On chip decoupling capacitance.

2-3° Create current models (noise source models) for clock tree, logic contents, memories, etc, and attach them to the power/ground netlist as follows.

- 1) For the global clock tree, if a clock driver component in the global clock exists in a coarse grid, attach it to the

power/ground RC tree in the grid, with load capacitance. If it is a gated clock, then a driver with control signal is used. (Fig. 8(b-1).)

- 2) For the lower clock tree, the clock buffer and clock leaves inside F/Fs in each coarse grid are modeled by using switching gate model. Also the random logic part in the grid is included in this switching gate model. (Fig. 8(b-2).) Details of this model are described later.
- 3) For parts such as a memory core, we can estimate power current by using a current source model. (Fig. 8 (b-3).) Details of this model are described later.

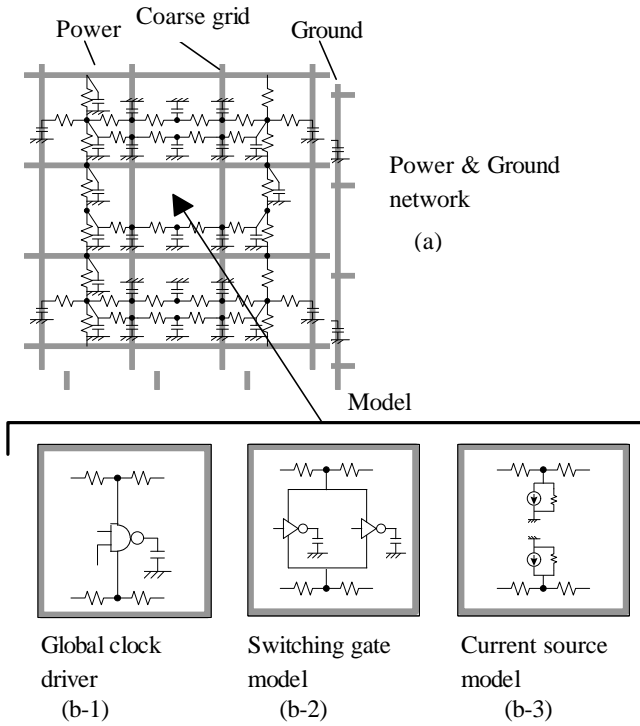


Fig.8 Network for AC power analysis (simplified)

switching gate model

This is a model consisting of a series of inverters or nand gates, to represent clock buffers, clock leaves and random logic part.

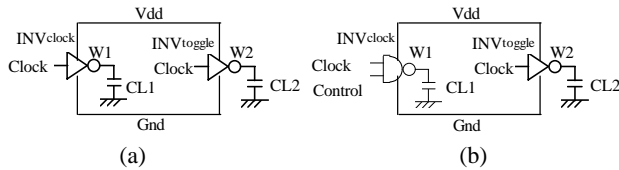


Fig.9 Switching gate model

INV_{clock} represents a clock related part in a grid. This corresponds to power $P_{clock}(i)$. The gate width $W1$, and the load capacitance $CL1$ of INV_{clock} are given by

$$W1 = \sum(W \text{ of Buffers in a grid}) + \sum(W \text{ of gates in F/Fs, in the grid, that switch at every clock}) \quad (12)$$

$$CL1 = \sum(\text{load of Buffers in a grid}) + \sum(\text{load of gates in F/Fs, in the grid, that switch at every clock}) \quad (13)$$

$CL1$ is equivalent to

$$\left\{ P_{clock}(i) / V_{dd}^2 * f \right\} / \left\{ A_i / A_{grid} \right\} \quad (14)$$

where V_{dd} is the power supply voltage, f is the clock frequency, and A_{grid} is the area of one coarse grid.

INV_{toggle} represents a logic part in a grid. The gate width $W2$, and the load capacitance $CL2$ are given by

$$W2 = Ng_i * (A_{grid} / A_i) * W_{gate_ave} * \alpha_i \quad (15)$$

$$CL2 = \left\{ P_{toggle}(i) / V_{dd}^2 * f \right\} / \left\{ A_i / A_{grid} \right\} \quad (16)$$

where α_i is the toggle rate for gates in $Block_i$, and W_{gate_ave} is the average gate width in $Block_i$.

More precisely, since not all gates in a grid switch at the same time, the inverter of size W_j ($j=1,2$) may be further partitioned into a series of inverters of size W_{j1}, W_{j2}, \dots , where $W_j = \sum(W_{j1} + W_{j2} + \dots)$, and the inverters of size W_{j1}, W_{j2}, \dots switch with delay $\delta_{j1}, \delta_{j2}, \dots$ compared to the clock cycle, to incorporate the effect that current peaks overlap with some delay.

Current source model

The current model in Fig. 10 takes into account the variation of device current caused by power voltage drop [4][5].

Let V_{cc0} be the ideal power supply voltage, and $I_{cc0}(t)$ be the power current waveform at V_{cc0} . Then the actual power current $I_{cc}(t)$, which corresponds to the actual fluctuating supply voltage $V_{cc}(t)$, is approximately given by

$$I_{cc}(t) = \frac{\partial I_{cc}}{\partial V_{cc}} * (V_{cc}(t) - V_{cc0}) + I_{cc0}(t) \quad (17)$$

Equation (17) means that the power current is expressed by an equivalent circuit with conductance $G (= \partial I_{cc} / \partial V_{cc})$, the power supply voltage V_{cc0} , and the current source $I_{cc0}(t)$ as shown in Fig.10(a). This circuit is equivalent to the circuit in Fig.8(b) with current source $(I_{cc0} - G * V_{cc0})$. G may be given, ignoring time dependency, by

$$G = \frac{I_{cc \max 0} - I_{cc \max 1}}{V_{cc0} - V_{cc1}} \quad (18)$$

where $I_{cc \max 0}$ is the peak current at V_{cc0} , $I_{cc \max 1}$ is the peak current at V_{cc1} , where V_{cc1} is the variation of say, 10% to the ideal voltage supply V_{cc0} .

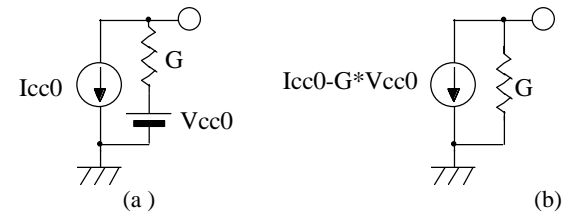


Fig.10 Current source model

This model is used for the memory core, for example. By measuring $I_{cc}(t)$ for the sense amplifier, decoder, address buffer, and together with conductance G , an equivalent current source model may be created.

2-4° Input the clock signal, and perform dynamic power simulation.

Using the step above, a number of power structures are analyzed in DC/AC mode. Electromigration, power voltage drop, and noise can be analyzed, and trade-off analysis between robustness and area is done.

Step 6. Clock skew/delay analysis considering power voltage drop

1. dynamic check

By supplying current in Step 5 to power/ground networks, clock skew and critical simulation may be performed by considering power voltage drop.

2. Static check

Static timing analysis considering power voltage drop may be performed as follows.

1. Calculate power voltage drop distribution in a chip as shown in Step 5.
2. Simulate delay-power supply voltage dependency for all cells in a library, and put them into the cell delay library.
3. By using this delay library, calculate all cell delays and interconnect delays depending on the voltage drop calculated in 1., then perform static timing check using this delays.

4. Discussion for accuracy

Fig. 11 shows the power breakdown of a microprocessor chip. A similar breakdown profile have been reported in [7][8][9].

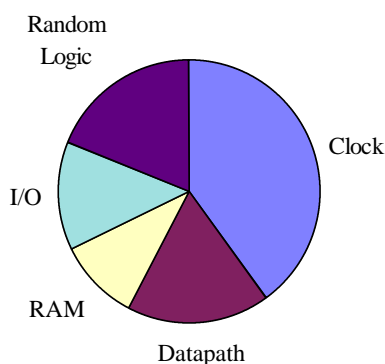


Fig. 11 Breakdown of power dissipation of a chip

The clock is the largest power consuming component. This includes the clock generators, clock drivers, the clock trees, and clock loading F/Fs.

The ratio between global clocks and local clocks of the power consumption of the clock portion varies with designs, from approximately 1:1 to 3:7. The lowest clock loading tends to be the largest component, because the number of F/Fs is large. This

strongly suggests that the distribution of clock loading elements (F/Fs) has a strong impact on overall chip signal integrity issues, such as power voltage drop, clock skew, and delay caused by power voltage drop.

Fig. 8 also shows that the datapath, RAM, and I/O are large power consuming components. They can be modeled as shown in Section 3.

By using a coarse grid approach and modeling techniques, dynamic transistor-level simulation can be performed in a short computational time. This is a significant advantage in terms of both accuracy and turn around cycle time of the analysis. This is in contrast to the conventional approach which has three major disadvantages:

The analysis is done late in the design cycle;

The analysis is applied using detailed extracted information of the nearly finished chip, making static analysis necessary (and dynamic analysis impossible); and

The static analysis ignores dynamic effects.

Stepwise refinement methodology

As seen in section 3, at an early design stage, statistical estimation is used. Later in the design this statistical estimation is re-visited and revised. For example, the number of gates and F/Fs could be estimated more precisely in gate-level logical design and optimization phase.

It should be also noted that the results of Level 1 analysis are used to set design guidelines, budgets, and constraints for use in the later portion of the design stage (Level 2 and Level 3). Those budgets/constraints are refined progressively as the chip design progress. Finally verification is done, which results in only minor fixes.

5. A Case Study

The design methodology has been applied and tested for several designs. The design flow from step 1 to step 6 in section3 has been applied to a mixed-analog-digital chip in Fig. 12. The objective of this case study is validate the design flow and the accuracy of coarse grid approach.

The power/ ground netlist model and clock tree netlist model for the coarse grid is illustrated in Fig. 13.

Simulated power dissipation and power/ground DC voltage drop showed good (less than 5% error) agreement with actual measurements. Power/ground voltage peak in dynamic power analysis showed less than 30% difference from the actual measurement. This power distribution profile is then used for clock skew analysis and static timing analysis. Critical path delay caused by power voltage drop increased from 2.27ns in a perfect power supply to 2.35ns, or 3.5%. This also reasonably matches the actual measurement. This worst critical path reported by static timing analysis does not exactly match the critical path on a chip, but if process variation is taken into account, the distribution of path delay can be thought to reasonably match the measurements. On-chip decoupling effect, and several alternatives for power grid widths were also analyzed to make design decisions.

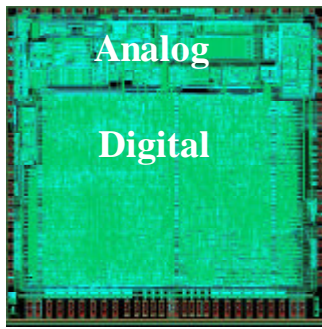


Fig. 12 A test case chip photo

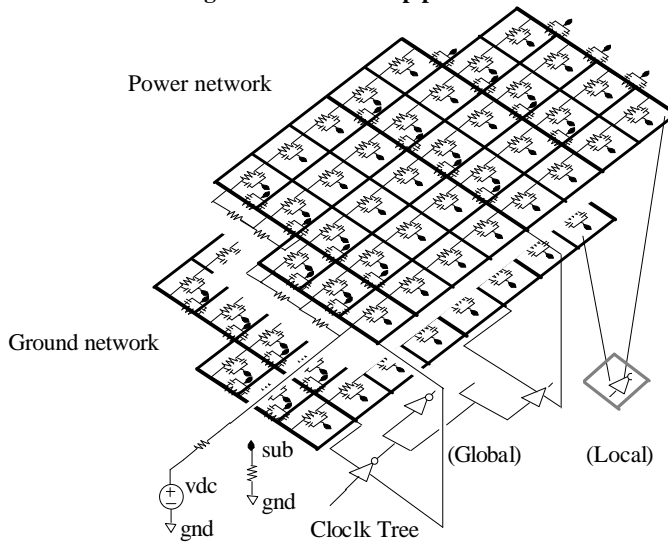


Fig. 13 Power/ground network and clock tree model construction

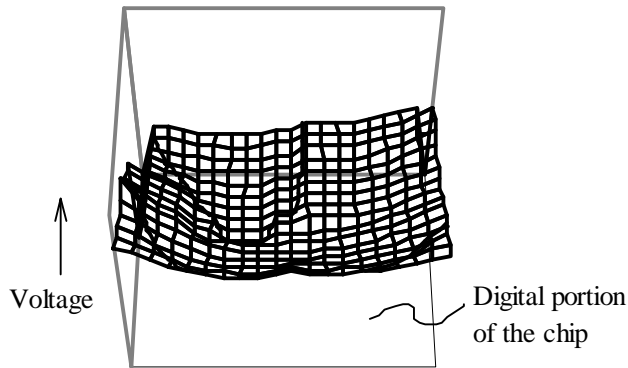


Fig. 14 Power distribution snapshot across the digital portion of the chip

6. Conclusion

This paper presents a methodology to analyze signal integrity such as power voltage drop and clock skew in the early stages of design. In the early design cycle, logic contents are not known, but global structure of power/ground and clock networks, function partitioning and early floorplan give reasonable accuracy for global optimization of the chip. A case study shows that this

methodology can be effectively applied to whole chip level power voltage drop analysis, and timing analysis considering power voltage drop.

Areas of future work involve fast noise analysis in the design planning stage, which includes substrate coupling noise.

Acknowledgment

The authors would like to thank many of our colleagues, especially Goichi Yokomizo and Kauhiko Hikasa, for many useful discussions, and for giving us a lot of insights.

References

- [1] Semiconductor Industry Association: "National Technology Roadmap for Semiconductors", (1997), Design & Test section, pp23-42.
- [2] G. Friedman, "Clock Distribution Networks in VLSI Circuits and Systems", IEEE PRESS, 1995.
- [3] G. Friedman, "High Performance Clock Distribution Networks", Kluwer Academic Publishers, 1997.
- [4] Miyama, G. Ykomizo, M. Iwabuchi, and M. Kinoshita: "An Efficient Logic/Circuit Mixed-Mode Simulation for Analysis of Power Supply Votage Function", Proc. of ASP-DAC, (1995) pp.366-371.
- [5] Ogawa, M. Iwabuchi, et. al.: USP: 5,481,484, "Mixed Mode Simulation Method and Simulator".
- [6] Abhijit Dharchoudhury, et al.: "Design and analysis of Power Distribution Networks in PowerPC™ Microprocessors", Proc. of DAC, (1998) pp738-743
- [7] Liu and C. Svensson: "Power Consumption Estimation in CMOS VLSI Chips", IEEE Journal of Solid-State Circuits, (1994) pp663-670
- [8] Michael K. Gowan, et. al.: "Power Consideration in the Design of the Alpha 21264 Microprocessor", Proc. of DAC, (1998) pp.726-731.
- [9] Vivek Tiwari, et. al.: "Reducing power in High-performance Microprocessors", Proc. of DAC, (1998) pp.732-737.