A Wakeup Rush Current and Charge-up Time Analysis Method for Programmable Power-Gating Designs

Kaijian Shi
Synopsys (Professional Services)
14911 Quorum Drive
Dallas, Texas, USA
+1 972 776 7490
Kaijian.Shi@synopsys.com

Jingsong Li
Synopsys
700 East Middlefield Rd.
Mountain View, CA, USA
+1 650 584 5360
jingsong@synopsys.com

ABSTRACT

Programmable power-gating structure is often implemented in advance low-power designs where switch cells are connected into a number of separate daisy chains which can be configured into a trickle charge chain and a main chain. The main chain can also be programmed with desired turn-on time to achieve short charge-up time while meeting max rush current constraints during wakeup. A chain latency annotation method is presented in this paper to model the chain configurations and programmable main-chain turn-on time of complex power-gating designs in the wakeup rush current and charge-up time analysis.

1 INTRODUCTION

Power-gating is one of the most effective standby-leakage reduction method recently developed [1]-[6] for sub-130nm low-power applications such as hand-held devices and battery operated equipments. In a power gating design, sleep transistors are used as switches to shut off power supplies to parts of a design in standby mode. In industrial power-gating designs, the sleep transistor is commonly custom designed as a switch cell which includes the sleep transistor optimized for drive and area efficiency and a repeater that drives the sleep transistor and relay the power-gating control signal. A good size power-gating design needs hundreds to thousands switch cells to provide sufficient drive current to the design. When the design is coming out of the sleep mode, the switch cells are turned on to supply power to the design. The simultaneous switching current of charging the design to a full power-on state can reach tens of Amps. Such large current surge will cause large IR-drop in the design because of the limited power supply in the battery operated devices. A practical method to control the wakeup rush current is to implement the switch cells in a daisy chain where the switch cells are driven by a buffer chain and turned on gradually in the delayed sequence. The wakeup rush current is controlled by the buffer delay which determines the delay of sequential turn-on of the switch cells. If the buffer delay was too short, the switch cells would be turned on close to simultaneously and cause large rush current. On the other hand, a buffer delay that is too long would result in excessive long wakeup latency due to slow charge of the design to full operation voltage. In industrial designs, the buffer delay is often constrained by the availability of the buffers in the cell library that the design implements. It is more often than not, that the delays of the buffers in the library are much smaller than desired. Adding more buffers trying to reach the desired delay will introduce power penalty on the buffers contradicting the purpose of low-power design and hence is not an option. To address the issues, advanced power-gating designs often implement complex power-gating structures where the switch cells are connected into a number of separate daisy chains of various lengths. The shortest chain has a few of the switch cells while the longest chain includes majority of the switch cells. These daisy chains are controlled by a power management unit which in turn can be programmed by either hardware wiring or application software for optimal wakeup latency reduction and rush current control. The diagram in Fig. 1 illustrates the complex power-gating structure.

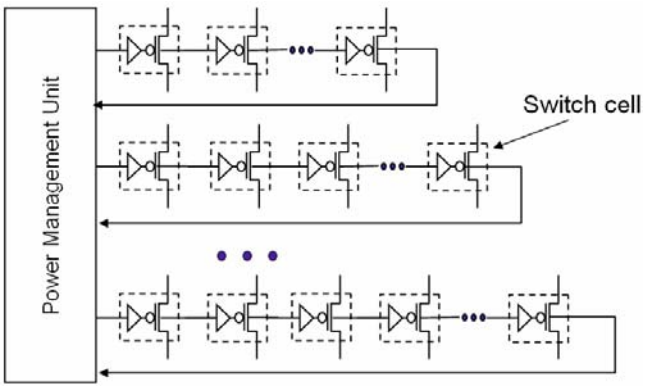


Figure 1. Programmable power-gating structure

To control rush current, one or two of the short chains are configured as a trickle charge chain to slowly charge the design and rest of the chains are configured as a main charge chain. At wakeup, the trickle chain is turned on first to trickle charge the design until the voltage on the virtual power network reaches close to Vdd, and then, the main chain is turned on to fully charge the design for normal operation. The rush current during the trickle charge is controlled by the number of switch cells configured in the trickle chain. Single shortest chain would result in smallest rush current limited by the saturation drain current of the sleep transistors in the chain. Weak switches can be implemented in the short chains to further reduce rush current. After the main chain fires, the rush current is mainly determined by the main chain turn-on time which is

programmable in the complex power-gating structure. The later the main chain fires, the smaller rush current because the virtual power network has been trickle charged close to Vdd by the time the main chain is turned on. The trade-off is the increase of the wakeup latency.

The configurable trickle chain and the programmable main chain turn-on time are the main advantage of the complex power-gating structure. The optimal combination of them generates an optimal charge pattern at wakeup resulting in minimum rush current while meeting wakeup latency constraints However, obtaining the optimal chain configuration and the main chain turn-on time requires post-layout transient simulation of wakeup rush current and latency of the design sweeping various combinations of the chain configurations and main chain turn-on time. Currently, commercial dynamic power-rail analysis tools handle only physically connected daisy chains and hence are not able to perform such programmable wakeup analysis. To address the issues and needs, we have developed a latency annotation method to model the chain configurations and programmable main-chain turn-on time of such complex power-gating designs in the wakeup analysis. In the remainder of the paper, we shall firstly describe the principle of wakeup rush current and latency analysis. Then we shall describe the developed chain-latency annotation method with pseudo-code descriptions. Next, we shall present an application example of wakeup analysis of a real industrial complex power-gating design using the method.

2 PRINCIPLE OF WAKEUP ANALYSIS

In the wakeup period, the switch cells in the daisy chains are turned on in the defined sequence to charge up the design which is commonly in reset state with all clocks stopped. The wakeup analysis is a time domain simulation of current rush and voltage ramp up during the charging up period. The design can be modeled in various ways in the wakeup analysis depending on the requirements of simulation time and accuracy. Such a simulation model is illustrated in the diagram in Fig. 2.

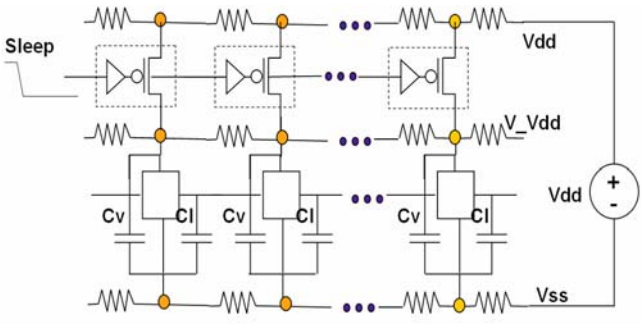


Figure 2. Wakeup analysis circuit

In this model, both permanent (Vdd) and virtual (V_Vdd) power networks are modeled by resistive networks which are extracted from the metal wires and vias in the power rails using parasitic extraction tools. The capacitance Cv in the diagram represents the decoupling capacitance on the power network including wire capacitance and inserted decoupling capacitors. The boxes in the diagram represent the logic cells and the Cl is the load capacitance extracted from the signal wires. Although the logic cells are not functionally switching during the wakeup, the cell leakage current and transient crowbar current have noticeable effect on the wakeup behavior. The load capacitance also affects the wakeup behavior through transistors in the cells. In wakeup analysis of very large designs, this simulation model is often further simplified to reduce the problem size in order to meet the simulation time and computer memory capacity constraints. The detailed of various wakeup simulation model reduction techniques are not the focus of the paper and hence not discussed further.

In the wakeup analysis, the simulation model is solved by transient analysis tools at discrete time points using numerical integration methods [7]-[9] where the capacitance is represented by its companion model and the wakeup model is represented by the modified nodal analysis [7]-[9] equation at a discrete time step. The principle of the transient analysis and numerical integration algorithms have been well documented in literature [7]-[9] and hence not repeated here. The sum of the current through all the switch cells forms the rush current in the wakeup period. The wakeup latency is obtained by the time, at which the voltage on the virtual power is charged up to a threshold of Vdd (e.g. 99% of Vdd).

3 LATENCY ANNOTATION METHOD

In the programmable power-gating designs, the switch chains are not physically connected. They are turned on by the power management unit based on the defined chain configuration and main-chain turn on time. Consequently, the simple daisy chain model as shown in Fig. 2 can no longer correctly model the wakeup behavior of the programmable power-gating design. To address this issue, we have developed a chain latency annotation method based on an event-controlled switch cell turn-on sequence model.

Simulations show that the switching power of those repeaters is the daisy chains (as shown in Fig. 3) has little effect on the rush current and charge-up latency in the wakeup period. Only the path delays of the repeaters from the power management unit (control registers) to the switches are important to the charge-up sequence. Therefore, we model the switch cell daisy chain as a group of switch cells each of which is turned on at certain time (event) determined by the path delay from the power management unit to the switch control pin. Thus, we map the daisy chain circuit into a number of switch-event-controlled switches as shown in the diagram in Fig. 3.

The event-controlled switch model maps physical daisy chains to an event-controlled power-on sequence. Based on this model, we further developed a latency annotation method to model various chain configurations and main-chain turn-on time. In the method, we annotate the daisy chain latency to the switch cells according to the chain configuration and main-chain turn-on time. The principle of the modeling is outlined by two case studies below:

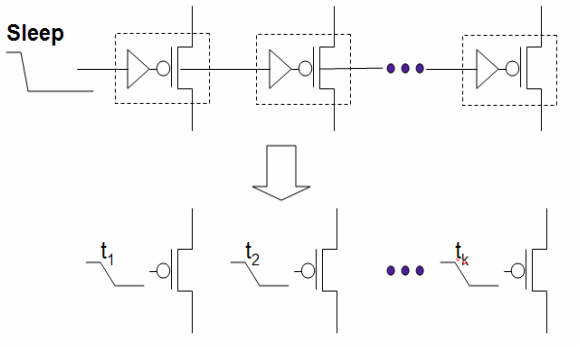


Figure 3. The switch daisy chain model

In the first case, the first and second short switch chains are configured as a pair of parallel chains to trickle charge the design at wakeup. The rest of the chains are also configured in parallel and turned on simultaneously at the programmable main-chain turn-on time to fully charge up the design. In this case, the events of the switches in the trickle chain are the time that the power-up control signal takes to propagate to the switches. We obtain the propagation time by calculating the path delay from the configuration registers in the power management unit to the switches using static timing analysis (STA) methods. For the switches in the main chain, their events are determined by both the path delays and the programmable main-chain turn-on time. We generate the switch events by annotating the main-chain turn-on time on the top of the calculated path delays to the switches. As the result, we are able to correctly model the turn-on sequence and time of the switches in the trickle and main chains according to the chain configuration and main-chain turn-on time in the wakeup analysis, in this case.

In the second case, the first and second short switch chains are configured as a consecutively connected chain to trickle charge the design at wakeup. The rest of the chains are also configured in series with the first chain being turned on at the main-chain turn-on time to fully charge up the design. In this case, we model the sequence and timing of the consecutively connected two trickle chains by calculating the chain latency of the first chain and annotating the latency on the top of the event time of the switch cells in the second chain. In this case, the chain latency includes the path delay to the last switch in the first chain. If the handshaking technique is implemented in the power-up control, the chain delay also includes the return path delay of the acknowledge signal path back to the power management unit. With this series chain latency annotation method, we are able to model the turn-on sequence and time of the switches in this series trickle chain configuration correctly correlating to the physical turn-on sequence and time. We apply the same chain latency annotation method to the switches in the main chain with an additional annotation of the main-chain turn-on time to the switches to align the switches events with the programmed main-chain turn-on time.

The methods described above for parallel and series chain configurations and for trickle and main charge chains are sufficient to handle all programmable combinations.

4 WAKEUP ANALYSIS OF PROGRAMMABLE POWER-GATING DESIGN BY THE CHAIN LATENCY ANNOTATION METHOD

We have implemented the latency annotation method in PrimeRail, an industrial EDA tool for wakeup analysis of the configurable power-gating designs. The principle of such an analysis is outlined below. In the wakeup analysis, we first generate the turn-on time (events) of the switch cells in those chains configured for trickle charge of the design by the latency annotation method described in the previous section, based on the chain selection and connection type (series vs. parallel) defined in the chain configuration. Then, we apply the same latency annotation method to the switch cells which are configured for main charge of the design, according to the main chain configuration and the main-chain turn-on time. At this point, all switch cells in the trickle and main charge chains are represented by the switch-event controlled models corresponding to the chain configurations and main-chain turn-on time. Next, we generated discrete time-domain analysis matrix form equations incorporating the switch-controlled models using the numerical integration methods [7]-[9] and solve the equations using time domain simulation solvers to accomplish transient analysis of wakeup rush current and latency time of the configurable power-gating design. Below are the pseudo-code descriptions of the wakeup analysis with the latency annotation method.

Programmable power-gating wakeup analysis

```
trickle_events = \
    latency_annotation(trickle_config, 0)
main_events = \
    latency_annotation(main_config, main_on_time)
wakeup_simulation(trickle_events, main_events)
```

Latency annotation

```
latency_annotation (chain_config, chain_on_time)
    latency = chain_on_time
    foreach chain_i in chain_config->chain_list
        foreach switch_i in chain_i
            calculate path delay (td) to switch_i
            event_t = td + latency
            event_list = append (switch_i, event_t)
        if chain_config->type is series_connection
            latency = latency + chain_delay(chain_i)
    return event_list
```

5 APPLICATION RESULTS

We have successfully applied the method to an industrial programmable power-gating design where the switch cells are implemented in a few of separate chains of different lengths and the main-chain turn-on time is also programmable. The rush current and ramp-up voltage curves in Fig. 4 show the results of the full-chip post-layout wakeup analysis (in worst-case PVT corner) of four programmable cases where two short chains are configured in parallel to trickle charge the design and the rest of the chains are configured also in parallel as a main chain to fully charge the design at four programmed main-chain turn-on time at 5ns, 10ns, 15ns and 20ns respectively. The red curves are the ramping voltage and the blue curves are the charge current during the wakeup.

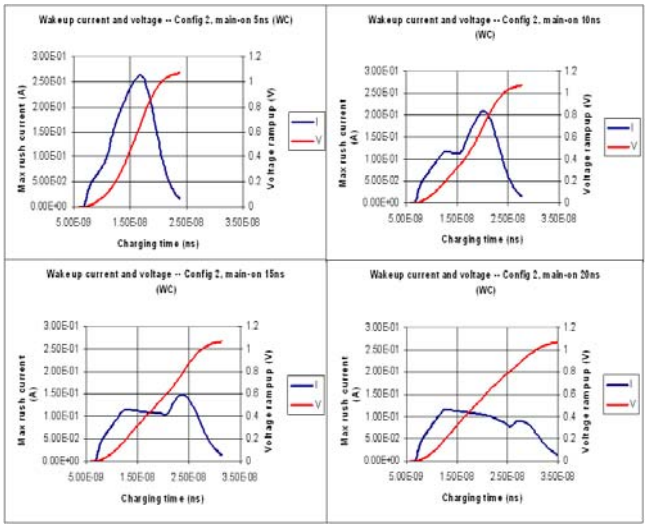


Figure 4. Wakeup current and voltage curves

The IV curves in the top-left graph correspond to the case of 5ns main-chain turn-on time. The curves show that the charge current increase with the number of the switches being turned on and reach the peak rush current of 0.25A at 17ns. Then current drops off as the design is being charged up, until 99% of Vdd at 24ns which is the charge-up latency. The charge current eventually dies down when the design is fully charged to Vdd. The IV curves are shifted from 0ns to 6ns corresponding to the propagation delay of the turn-on control signal form the configuration registers to the switch cells through repeaters in the layout.

In the case of 10ns main-chain on time, the IV curves in the top-right graph show that the peak rush current is reduced to 0.2A due to longer trickle charge period and the charge-up time goes up to 28ns. The current levels off in the middle of the curve corresponding to the constant current charge period when all the switch cells in the trickle chain have turned on while rest of the switches are not yet on until the time the main-chain is turned on and the current starts to increase again.

Increasing the main-chain turn-on time to 15ns reduces the peak rush current as shown in the curves in bottom-left graph, because the design is charged more close to Vdd when the main chain fires. However, further increase the main-chain turn-on time longer than 20ns does not reduce further the peak rush current, as show by the IV curves in the bottom-right graph, because the design has already been charged up before main chain fires. In this case, the peak rush current becomes determined purely by the trickle chain and at lowest possible level of 0.12A.

6 SUMMARY

Programmable power-gating technique becomes popular in industrial power-gating designs where switch cells are connected into a number of daisy chains which can be configured into a trickle charge chain and a main chain. The main chain can also be programmed with desired turn-on time to achieve low peak rush current while meeting charge-up time constraints during wakeup. Current wakeup analysis EDA tools do not correctly analyze such programmable power-gating designs. We have developed a latency annotation method to model the chain configurations and programmable main-chain turn-on time. With the method, wakeup behavior of the designs in various chain configurations and main-chain turn-on time is analyzed correctly correlating with the wakeup charge patterns determined by the programmed configurations. The method has been applied successfully to an industrial power-gating design with correct results. The method has been implemented in PrimeRail – a commercial EDA tool for static and dynamic IR-drop analysis.

7 REFERENCES

[1] Kaushik Roy, Saibal Mukhopadhyay, and Hamid Mahmoodi-meimand, “Leakage current mechanism and leakage reduction techniques in deep-submicrometer CMOS circuits”, Proc. IEEE Vol. 91, no. 2, Feb. 2003

[2] Dongwoo Lee, David Blaauw, and Dennis Sylvester, “Gate oxide leakage current analysis and reduction for VLSI circuits”, - IEEE Trans. VLSI, Vol. 12, No. 2, Feb. 2004

[3] M. Powell, S.-H Yang, et. al. “Gated-Vdd: A circuit technique to reduce leakage in deep-submicron cache memories”, in Proc. Int. Symp. Low Power Electronics Design, 2000

[4] Satoshi Shigematsu et. al., “A 1-V high-speed MTCMOS circuit scheme for power-down application circuits”, IEEE J. Solid-State Circuits, vol. 32, no. 6, June, 1997

[5] Benton H Calhoun, Frank A Honore and Anantha P Chandrakasan, “A leakage reduction methodology for distributed MTCMOS”, IEEE J. Solid-State Circuits, vol. 39, no. 5, May, 2004, pp. 818-826

[6] Changbo Long and Lei He, “Distributed sleep transistor network for power reduction”, Proc. IEEE/ACM Design Automation Conference, 2003

[7] Jan Ogrodzki. "Circuit Simulation Methods and Algorithms", CRC Press, Inc. 1994

[8] Jiri Vlach, Kishore Singhai," Computer Methods for Circuit Analysis and Design”, Kluwer Academic Pub., 2003

[9] Chua, L. O. and P. Lin "Computer-aided Analysis of Electronic Circuits: Algorithms and Computational Techniques", Prentice-Hall, Englewood Cliffs, NJ., 1975