Power Gating for Lower Power Designs

Xung Nham
George Mason University
xnham@gmu.edu
Scholarly Paper Advisor: Dr. Jens-Peter Kaps

Abstract

As process technologies have shrunken, leakage current has become an increasing portion of power loss on modern processors. The technique of power gating can solve most leakage instances. However, because the method requires design tradeoffs to implement, there are many tracks of improvement that can be considered when implementing power gating. This paper presents various optimizations that have been implemented or proposed to achieve more optimal power savings. Those methods include distributed sleep transistor networks, improving gate sizing, and multiple sleep modes.

Introduction

As process technologies have scaled down in size, the supply voltage has been reduced due to the thinning of the gate oxide layer and the shrinking of gate dimensions. This reduction in supply voltage has an effect of lowering power consumption during dynamic switching. This can be observed in equations 1 and 2 which are equations for MOSFET current in the linear and saturation regions respectively.

$$I_D = \mu_n C_{ox} \frac{W}{L}\left(V_{GS} - V_{th} - \frac{V_{DS}}{2}\right)V_{DS} \qquad (1)$$

$$I_D = \frac{\mu_n C_{ox}}{2}\frac{W}{L}(V_{GS} - V_{th})^2(1+\lambda V_{DS}) \qquad (2)$$

μ_n is the charge-carrier effective mobility, W and L are the gate width and length, C_{ox} is the gate oxide capacitance, V_{th} is the threshold voltage, V_{GS} is the gate to source voltage, V_{DS} is the drain to source voltage, and λ is a channel length modulation parameter. A decrease to supply voltage results in a decrease of drain current, thus dynamic switching power consumption.

The propagation delay of a CMOS gate can be approximated by equation 3.

$$T_{pd} \propto \frac{C_L V_{dd}}{(V_{dd} - V_{th})^{\alpha}} \qquad (3)$$

C_L is the load capacitance, V_{th} is the threshold voltage, and α is the velocity saturation index for modeling short channel effects. So while downward voltage scaling reduces power consumption during dynamic switching, it increases delay of the switching. To counteract the resulting delay increases, threshold voltages are lowered. Sub-threshold current is approximated by equation 4 where I_{D0} is the current when V_{GS}=V_{th}, V_T is kT/q which is a voltage that is a function of temperature, and n is a slope factor equal to 1+C_D/C_{ox}, with C_D the capacitance of the depletion layer.

$$I_D \approx I_{D0} e^{\frac{V_{GS} - V_{th}}{nV_T}} \qquad (4)$$

Thus as threshold voltage decreases, it causes sub-threshold leakage current to increase.

Figure 1 depicts a first order analysis of leakage power trends [12]. Without a way to reduce sub-threshold leakage, leakage power would grow to be the largest component of power consumption. This necessitates the need to develop techniques to reduce static power consumption during periods of idleness.

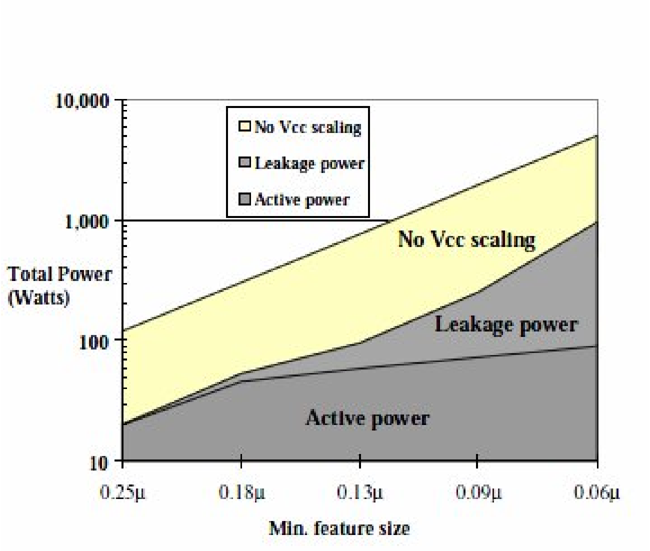


Figure 1: First order analysis of leakage power trends

One such technique is known as power gating where a sleep transistor is placed between the ground and a virtual ground (known as a footer switch), or for the PMOS version, placed between the V_{dd} and a virtual V_{dd} (header switch). The sleep transistors turn off during a sleep mode to cut-off the leakage path of the device. These sleep transistors are high threshold voltage transistors to provide low leakage. This technique provides a substantial reduction in leakage. However, the addition of sleep transistors cannot be without impacting performance, area, signal/power integrity, and various other effects. Powering down the blocks can be accomplished either by software or hardware. Driver software can schedule the power down operations or a dedicated power management controller can perform the task.

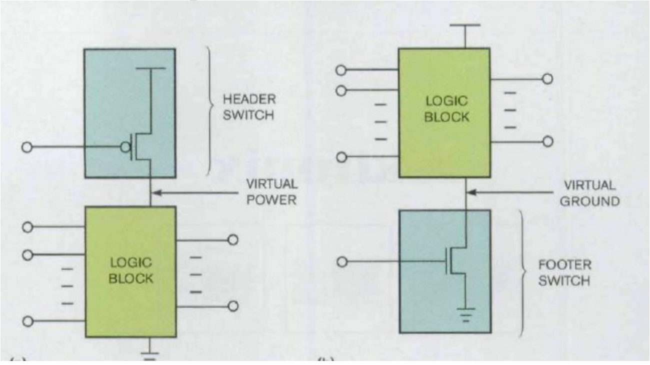


Figure 2: PMOS and NMOS versions of sleep transistors

The sleep transistors need to be optimized so the benefits of leakage reduction during sleep modes can outweigh the penalties of power and area introduced by them. These design decisions must be performed during the design phase as tool automation is not sufficient to create the most optimal implementations [11].

Fine vs. Course Grain Designs

Fine grain implementations consist of a sleep transistor being inserted into every standard cell. Figure 3 depicts an example cell-based NAND gate with sleep transistor. The cell has a weak pull-up/down transistor to prevent floating output during the sleep mode. This is prevents short circuit current in active cells that may be connected to the output of the sleep cell. The pull-up/down transistor remains in OFF state in normal operation mode. The design only allows one isolation state which is “1” in footer switch version and “0” in the header switch version.

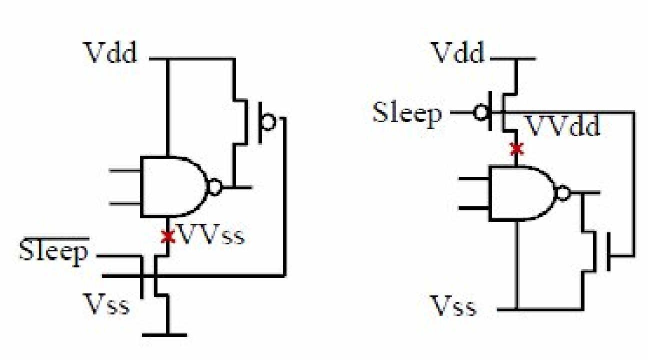


Figure 3: Footer and header cell-based sleep transistor implementation of a NAND gate

The advantages are that the high level of granularity allow for short power-on time. The cells can also be integrated into designs using existing standard cell synthesis and design tool sets. Cell-based sleep transistor designs however incur a large area overhead and increased routing complexity due to the sleep signals. The sleep transistor is also sized for the worst case scenario – that the gate will switch during every clock cycle because the switching frequency is unknown at the cell level. The cell is subject to more sensitivity to process, voltage, and temperature (PVT) variations, because the built-in sleep transistor is subject to PVT variations which causes IR-drop variations.

In course grain implementations, the power-gating transistor is part of the power distribution network rather than the cell. The circuit is partitioned into clusters of logic and each cluster is connected to a local sleep transistor. This also significantly reduces the area overhead. A variation of this clustered implementation is a distributed sleep transistor network (DSTN), where the sleep transistors of these clusters connect to a shared virtual ground or virtual power supply. This assumes that the clusters share the same sleep signal.

The advantages of a DSTN are that current from one cluster can flow through all sleep transistors. This allows the total sleep transistor size to be smaller than the total area of sleep transistors in a clustered implementation. Consider the case where every cluster consisted of only one gate. The clustered version would need sleep transistors sized for the peak current of every gate whereas the DSTN version would only need sleep transistors sized for the switching current during the worse case input vector. The sharing of sleep transistors tend to balance IR-drop occurrences and makes the design less sensitive to PVT variations. Most industry designs implementing power-gating employ the distributed sleep transistor implementation. Since the most benefit of power gating is through course grain implementations, the rest of the paper will talk about course grain concerns.

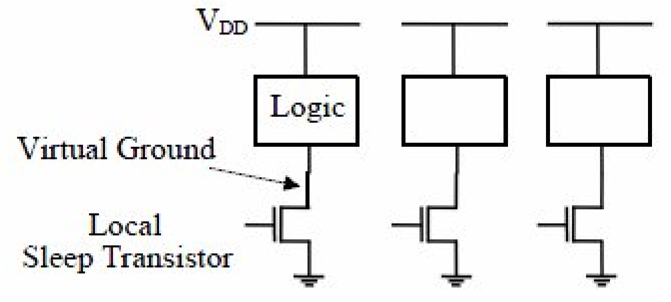


Figure 4: Cluster based implementation

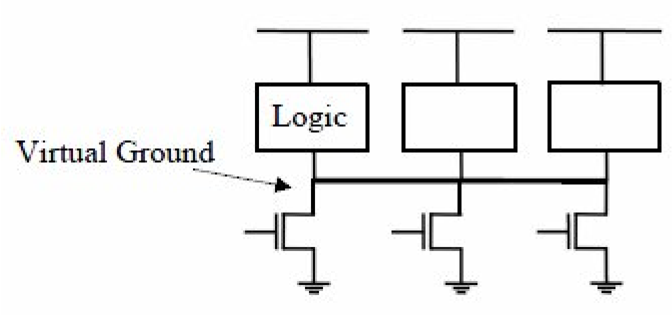


Figure 5: DSTN implementation

Sleep Transistor Sizing

Without a sleep transistor, the propagation delay of a CMOS gate can be approximated by

$$T_{pd} \propto \frac{C_L V_{dd}}{(V_{dd} - V_{tL})^{\alpha}} \qquad (5)$$

C_L is the load capacitance, V_{tL} is the threshold voltage of the lower threshold voltage components, and α is the velocity saturation index for modeling short channel effects. The addition of a sleep transistor is described in equation (2) – the sleep transistor increases the propagation delay as a function of the voltage drop over the sleep transistor (V_{st}).

$$T_{pd-MT} \propto \frac{C_L V_{dd}}{(V_{dd} - V_{st} - V_{tL})^{\alpha}} \qquad (6)$$

The sleep transistor must be sized to handle the maximum instantaneous current (MIC) at any given time. The transistor must be big enough such that there is no significant IR-drop due to the sleep transistor as defined by design constraints and the relation of IR-drop to propagation delay described by equation 6. Yet the transistor must be as small as possible to maximize the leakage reduction that the transistor can provide. Generally, the rule of thumb is 3 times the switching capacitance for the gate size. Transistor sizing depends on the overall switching current of the cluster at any given time. For a DSTN implementation, since only a fraction of circuits switch at any point of time, sleep transistors can be smaller than the aggregate size of sleep transistors for a cell-based implementation. The DSTN greatly complicates the problem of sizing the transistor because to maximize the leakage reduction requires minimizing voltage drops within constraints and considering the current distribution of the network.

When DSTNs were originally proposed [9], a suggested method for sizing was to use a weighted MIC of the circuit (1+ β)*MIC(CKT) for the total area of all sleep transistors with the area proportioned to each sleep transistor based on the MIC of each cluster. β is an empirical number between 0.05 and 0.5 obtained through trial-and-error of whether all IR-drop constraints were met. This method however doesn't account for the current distribution among sleep transistors because it only considers the maximum current of the entire circuit rather than minimizing each sleep transistor based on its timing constraints.

Another method for sizing based on timing driven constraints is proposed by [3]. The method models the DSTN as a resistance network such as Figure 6 where R_{ST} is the resistance for a sleep transistor and R_V is the resistance for the virtual ground lines. The currents through the sleep transistors are at a constant ratio as derived from Kirchhoff's current law and Ohm's law. Assuming $R_{ST} = (R_{ST1}, R_{ST2}, R_{ST3}, R_{ST4}) = (8, 9, 8, 10)$ and $R_V = (R_{V1}, R_{V2}, R_{V3}) = (1, 2, 2)$, then the current source of the first cluster, $I_{clus1} = \{0.38I_{clus1}, 0.27I_{clus1}, 0.21I_{clus1}, 0.14I_{clus1}\}$.

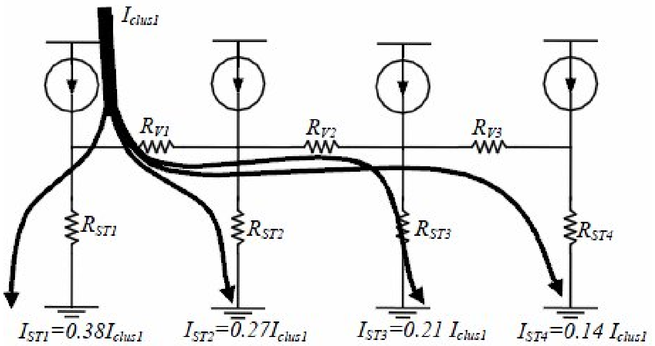


Figure 6: Current discharging with constant ratios

Repeating this for the other clusters, we can obtain a discharge matrix:

$$I_{ST} = \Phi \bullet I_{CLUS}$$

$$\begin{bmatrix} I_{ST1} \\ I_{ST2} \\ I_{ST3} \\ I_{ST4} \end{bmatrix} = \begin{bmatrix} 0.38 & 0.30 & 0.21 & 0.18 \\ 0.27 & 0.30 & 0.21 & 0.18 \\ 0.21 & 0.24 & 0.35 & 0.28 \\ 0.14 & 0.16 & 0.23 & 0.36 \end{bmatrix} \begin{bmatrix} I_{clus1} \\ I_{clus2} \\ I_{clus3} \\ I_{clus4} \end{bmatrix}$$

I_{CLUS} however depends on the input vector of the circuit and requires matrix calculations of all input permutations to find a maximum I_{ST}. Sizing the transistors would also require calculating the discharge matrix and repeating the permutations. Thus to estimate a maximum I_{STi}, the following method is used.

MIC(ST) Upper Bound Estimation

1. Find the I_{clusi} with the largest corresponding r
2. Maximize the I_{clusi} found in step 1 under all the MIC constraints
3. Substitute I_{clusi} in all equations with the maximum value I_{clusi}^*
4. If not all the I_{clusi}^* has been calculated, goto step 1
5. Substitute all I_{clusi} in the objective function with I_{clusi}^* to get MIC(ST)
6. Return MIC(ST)

The method uses the MIC of the clusters as constraints for I_{CLUS}. It also assumes a heuristic for finding the MICs of the clusters such as the one described in [5]. Thus, the maximum instantaneous currents {MIC(I_{clus1}), MIC(I_{clus2}), MIC(I_{clus3}), MIC(I_{clus4})} for all of the clusters as well different combinations of MICs {MIC(I_{clus1}, I_{clus2}), MIC(I_{clus3}, I_{clus4}), etc} are known at the beginning. The I_{clusi} are subject to the following constraints:

$$I_{clus1} + I_{clus2} + I_{clus3} + I_{clus4} \leq MIC(I_{clus1}, I_{clus2}, I_{clus3}, I_{clus4})$$

$$I_{clus1} + I_{clus2} \leq MIC(I_{clus1}, I_{clus2})$$

$$I_{clus3} + I_{clus4} \leq MIC(I_{clus3}, I_{clus4})$$

$$I_{clusi} \leq MIC(I_{clusi})$$

The method deals with one transistor at a time and applies the constraints until an estimated upper bound for all I_{STi} have been calculated.

With a method for estimating MIC(ST), the following method is used to size the transistors. The algorithm starts with the smallest sleep transistor available in the library. Then an estimation for MIC(ST) is made, which is the largest current flowing through the sleep transistors. Then, the sleep transistor with the worst voltage drop is enlarged to meet voltage drop constraints. The discharge matrix is updated according to the new transistor size. The steps are repeated until all sleep transistors meet voltage drops constraints.

```
Resizing Heuristic
1. Initialize sleep transistors to smallest size
2. Calculate discharging matrix $\Phi$
3. Update the sleep transistor MICs and voltage
   drops
4. If all voltage drops meet the constraints, goto
   step 6
5. Resize the ST with the worst voltage drop,
   goto step 2
6. Return size of all STs
```

Multiple Sleep Modes

A means of increasing the effectiveness of power gating as proposed by [8] is to introduce multiple intermediate sleep modes. Power gating reduces leakage because when the sleep transistor is off, the virtual ground rail charges up to a steady state value close to V_{DD}. However, to switch out of the sleep mode, the virtual ground has to discharge through the sleep transistor. This wake-up latency and power penalty limits the frequency that a cluster can go in and out of sleep. Multiple sleep modes allow trading-off wake up penalty for leakage savings.

The leakage savings and wake-up penalty are a function of the steady state voltage of the virtual ground [8]. By biasing the footer transistor in the weak inversion region, the steady state virtual ground voltage (V_{GND}) is derived as the following relation:

$$V_{GND} = \frac{-V_G + S_S \log_{10}\left(\frac{W_{CIRCUIT}}{W_{FOOTER}}\right) + \left(V_{THF} - V_{THC}\right) + \eta V_{DD}}{2\eta} \qquad (7)$$

V_{THF} and V_{THC} are the threshold voltages of the footer and the logic circuit. W_{FOOTER} and $W_{CIRCUIT}$ are an equivalent transistor width of the footer and the cluster's logic circuit. η Is the drain induced barrier lowering (DIBL) coefficient and S_S is the subthreshold slope. The equation shows that the virtual ground voltage is a linear function of the footer gate voltage (V_G) with a negative slope. This allows us to control the virtual ground voltage by biasing the footer gate voltage to different levels.

Controlling the virtual ground voltage provides the ability to control the trade-offs between leakage savings and overhead due to waking up. Representing the leakage current of a circuit in the active mode by I_{active}, the leakage savings can be described by equation 8:

$$\frac{I_{sleep}}{I_{active}} = 10^{-\left(\frac{\eta(V_{DD}-V_{GND})}{S_S}\right)} \qquad (8)$$

The equation shows that a higher virtual ground results in higher leakage savings. However, wakeup time and wakeup energy are described by equations 9. $C_{CIRCUIT}$ is the total capacitance of the cluster's logic circuit and $I_{ON,Footer}$ is the current of the footer transistor after it has been turned on to wake up the circuit.

$$T_{Wakeup} = \frac{C_{CIRCUIT} V_{GND}}{I_{ON,Footer}} \qquad E_{Wakeup} = \frac{1}{2} C_{CIRCUIT} V_{GND}^{2} \quad (9)$$

So using a higher virtual ground to achieve a high leakage reduction results in higher wakeup latency and energy penalty and vice versa. This leads to the conclusion that biasing the footer gate to the lowest voltage would yield a mode of operation with the most leakage savings at a cost of the most wakeup delay and energy and vice versa, the highest voltage for the footer gate would yield a mode with the least leakage savings while costing the least wakeup delay and energy.

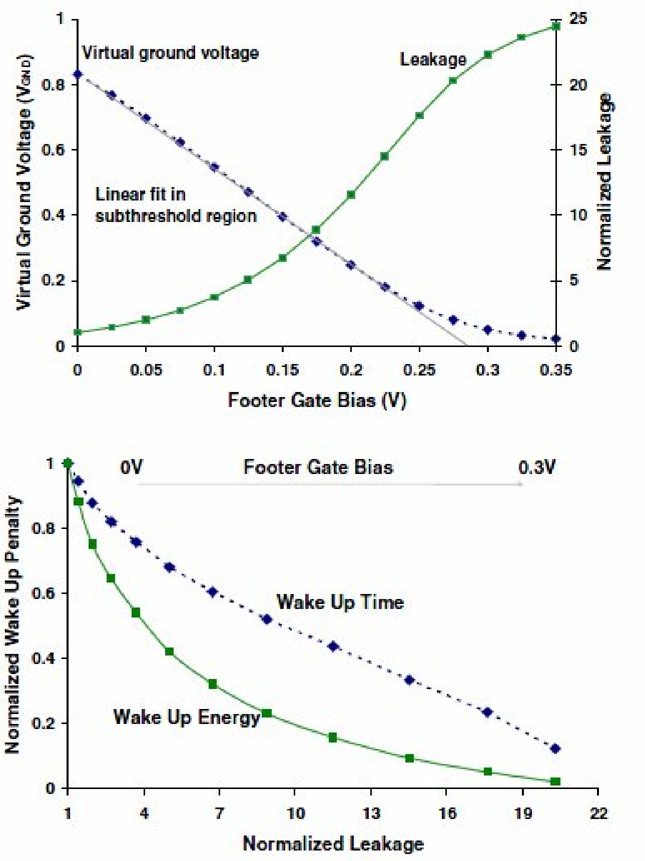
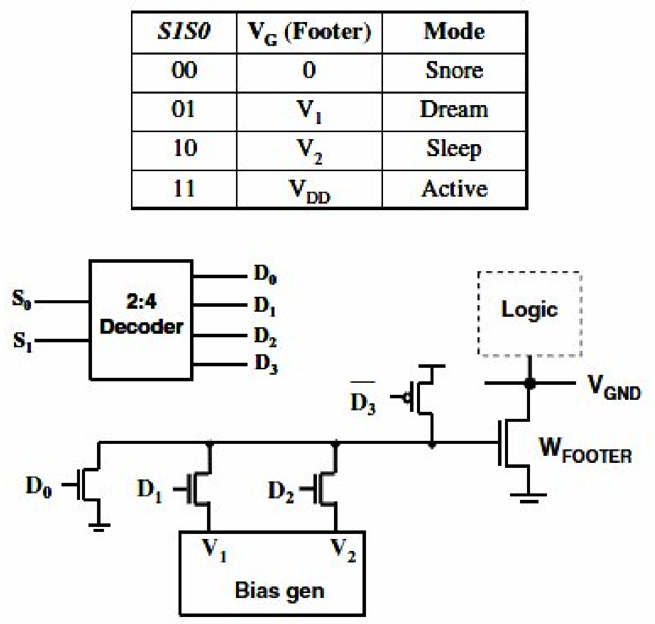


Figure 7: Effect of footer gate voltage on virtual ground and leakage vs wake-up trade-off

The charts of Figure 7 graph the relations of equations 7, 8, and 9. They depict the intermediate leakage savings, and consequent wakeup time and energy if the footer gate was biased to intermediate voltage levels. Using these intermediate states of footer gate voltages as intermediate sleep modes gives us an arbitrary number of sleep modes with different overhead costs. Multiple sleep modes allow trading-off leakage saving for wake up latency and overhead energy which would allow for more opportunities to enter an intermediate sleep mode without incurring additional performance latency.

A proposed circuit for using multiple sleep modes is depicted in Figure 8. The circuit has four modes - Snore, Dream, Sleep, and Active that are controlled by a two-bit selection signal.



S1S0	V_G (Footer)	Mode
00	0	Snore
01	V_1	Dream
10	V_2	Sleep
11	V_{DD}	Active

Figure 8: Multiple sleep mode circuit

Applying the intermediate sleep modes to empirical runs where instances of registered data had remained constant for less than the conventional wakeup latency but enough for the intermediate modes results in additional instances of sleep as depicted in Figure 9.

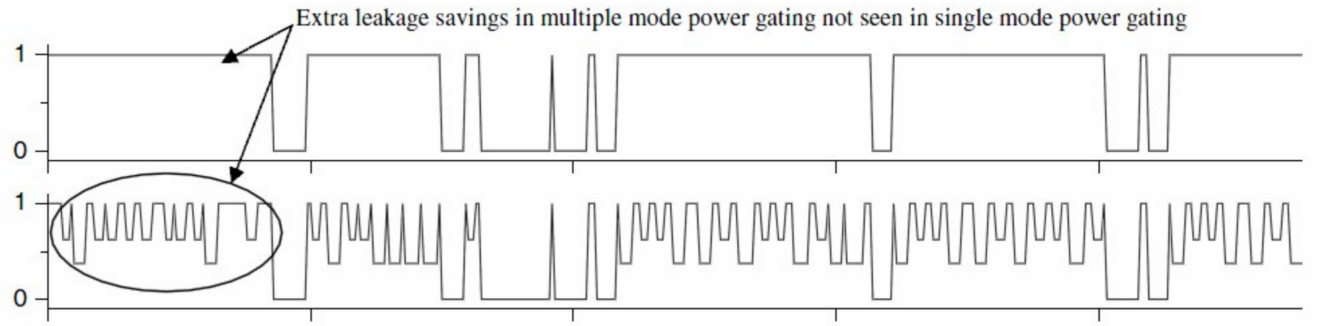


Figure 9: Sleep mode timeline of single and multiple mode power gating

The top chart shows the normalized voltage over time of a processor run utilizing a single sleep mode. The bottom chart shows the normalized voltage over time of the processor run utilizing multiple sleep modes. Because the intermediate sleep modes require less wakeup latency, the processor is able to enter into an intermediate sleep mode to save power at instances that it was not able to in the single sleep mode run.

Multiple sleep modes require a more sophisticated power-gating control. To maximize the benefit of intermediate modes, deterministic or correct estimation of required wake-up latency during application runtime is needed.

Conclusions

Moving forward with the need use smaller process technologies for faster designs has necessitated the development of power gating to curb leakage current. The largest chips have taken improving sub-threshold leakage current a priority. Because the addition of power gating sleep transistors cannot be without impact to performance, area, and signal/power integrity, power gating remains a highly researched topic. The methods presented are only a subset of the different ways of achieving the most leakage current reduction or mitigating the adverse effects of power gating.

For some methods, such as gate sizing, tool automation can provide this benefit through well defined, concise heuristics. Other concerns, such as DSTN implementation and multiple sleep modes require attention during the layout and architecture design stages of development. The trend has been to impose more and more of these power gating considerations onto the designer, so in the future, the most power efficient designs will require the most power gating considerations.

References

The following papers were either referenced directly or were used for general background knowledge on specific topics.

[1] Arindam Mukherjee, Malgorzata Marek-Sadowska, "Clock and Power Gating with Timing Closure," IEEE Design and Test of Computers, vol. 20, no. 3, pp. 32-39, May/June 2003.

[2] De-Shiuan Chiou, Da-Cheng Juan, Yu-Ting Chen, Shih-Chieh Chang, “Fine-grained sleep transistor sizing algorithm for leakage power minimization,” Proceedings of the 44th annual conference on Design automation, June 04-08, 2007, San Diego, California.

[3] De-Shiuan Chiou, Shih-Hsin Chen, Shih-Chieh Chang, Chingwei Yeh, “Timing driven power gating,” Proceedings of the 43rd annual conference on Design automation, July 24-28, 2006, San Francisco, CA, USA.

[4] Ehsan Pakbaznia , Farzan Fallah , Massoud Pedram, “Charge recycling in MTCMOS circuits: concept and analysis,” Proceedings of the 43rd annual conference on Design automation, July 24-28, 2006, San Francisco, CA, USA.

[5] Hsieh, C. T., Lin, J. C., and Chang, S. C., “A Vectorless Estimation of Maximum Instantaneous Current for Sequential Circuits,” Proceedings of ICCAD, pp. 537-540, 2004.

[6] Hyung-Ock Kim , Youngsoo Shin , Hyuk Kim , Iksoo Eo, “Physical design methodology of power gating circuits for standard-cell-based design,” Proceedings of the 43rd annual conference on Design automation, July 24-28, 2006, San Francisco, CA, USA.

[7] Kaijian Shi , David Howard, “Challenges in sleep transistor design and implementation in low-power designs,” Proceedings of the 43rd annual conference on Design automation, July 24-28, 2006, San Francisco, CA, USA.

[8] Kanak Agarwal , Kevin Nowka , Harmander Deogun , Dennis Sylvester, “Power Gating with Multiple Sleep Modes,” Proceedings of the 7th International Symposium on Quality Electronic Design, p.633-637, March 27-29, 2006.

[9] Long, C., and He, L., “Distributed Sleep Transistor Network for Power Reduction,” Proceedings of the 40th DAC, pp. 181-186, 2003.

[10] Pietro Babighian , Luca Benini , Enrico Macii, “Sizing and Characterization of Leakage-Control Cells for Layout-Aware Distributed Power-Gating,” Proceedings of the conference on Design, automation and test in Europe, p.10720, February 16-20, 2004.

[11] Santarini, Michael, “Taking a bite out of power: techniques for low-power-asic design," *EDN*, May 24, 2007, pg 46-64.

[12] Tiwari, V., Singh, D., Rajgopal, S., Mehta, G., Patel, R., and Baez, F. “Reducing power in high-performance microprocessors,” Proceedings of the 35th Annual Conference on Design Automation, June 15–19, 1998, San Francisco, CA.

[13] Yi-Ping You , Chung-Wen Huang , Jenq Kuen Lee, “Compilation for compact power-gating controls,” ACM Transactions on Design Automation of Electronic Systems (TODAES), v.12 n.4, p.51-es, September 2007.

[14] You, Y.-P., Lee, C., and Lee, J. K. 2002. “Compiler analysis and supports for leakage power reduction on microprocessors.” In Proceedings of the International Workshop on Languages and Compilers for Parallel Computing (LCPC) (Washington, DC), 63-73.