Power Gating with Multiple Sleep Modes

Kanak Agarwal, Harmander Deogun[Ŧ], Dennis Sylvester[Ŧ], Kevin Nowka

IBM Research, Austin, TX, 78758
[Ŧ]University of Michigan, Ann Arbor, MI, 48109

kba@us.ibm.com, hdeogun@umich.edu, dennis@eecs.umich.edu, nowka@us.ibm.com

Abstract – This paper describes a power gating technique with multiple sleep modes where each mode represents a trade-off between wake-up overhead and leakage savings. We show that high wake-up latency and wake-up power penalty of traditional power gating limits its application to large stretches of inactivity. Our simulations and data traces show that multiple sleep mode capability provides an extra 17% reduction in overall leakage as compared to single mode gating. The multiple modes can be designed to allow state-retentive modes. The results on benchmarks show that a single state-retentive mode can reduce leakage by 19% while preserving state of the circuit.

1. Introduction

The scaling of process technologies to nanometer regime has resulted in a rapid increase in leakage power dissipation [1,2]. Hence, it has become extremely important to develop design techniques to reduce static power dissipation during periods of inactivity. The power reduction must be achieved without trading-off performance which makes it harder to reduce leakage during normal (runtime) operation. On the other hand, there are several techniques for reducing leakage power in *sleep* or *standby* mode. Power gating is one such well known technique where a *sleep transistor* is added between actual ground rail and circuit ground (called *virtual ground*) [3,4]. This device is turned-off in the sleep mode to cut-off the leakage path. It has been shown that this technique provides a substantial reduction in leakage at a minimal impact on performance [5,6,7].

Power gating results in a reduction in leakage because when the sleep transistor is off, the virtual ground rail charges up to a steady state value close to V_{DD}. However, it also has a drawback that while switching back to the *active* mode from the sleep mode, the virtual ground rail takes a long time to discharge through the sleep transistor. This results in a significant wake-up latency and wake-up power penalty and limits overall leakage savings by limiting how often a logic block can go in and out of the sleep mode. Thus, it seems prudent to have multiple sleep modes that trade-off wake up penalty for leakage savings. During a stretch of inactivity, the processor can go into one of the intermediate sleep modes as determined by the wake-up overhead and save power without degrading performance. Unlike conventional power gating, the multiple sleep mode capability also provides an option of state-retentive mode to enable power savings during inactive periods while preserving the state of the circuit.

The concept of more than one low power modes is not entirely new. A circuit for intermediate power saving mode was proposed in [8]. In this reference, the authors propose using a PMOS device in parallel with the NMOS footer. In the intermediate mode, the PMOS device is turned-on while the NMOS footer is off. This holds the virtual ground rail potential at the threshold voltage of the PFET. However, this approach allows only one intermediate mode and the virtual ground rail potential of the intermediate mode is set by the threshold voltage of the PMOS device and cannot be arbitrarily controlled.

This paper describes a power gating circuit that supports operation in multiple sleep modes. Each mode represents a trade-off between wake-up overhead and leakage savings with larger wake up penalty modes resulting in higher leakage savings and *vice versa*. Fundamentally, both the leakage savings and the wake-up penalty of a sleep mode depend on the steady state potential of the virtual ground rail. A higher value of the steady state virtual ground potential (V_{GND}) results in higher leakage savings because it reduces the voltage across the logic circuit. At the same time, it also results in higher wake-up penalty because more charge has to discharge through the footer device. Hence, a trade-off between wake-up penalty and leakage saving in a sleep mode can be obtained by controlling the steady state V_{GND} potential in the sleep mode. This is the fundamental idea behind this approach.

The proposed circuit controls the steady state virtual ground rail potential by controlling the gate voltage of the footer device. In the sleep modes, the footer device is always biased in the weak inversion region. We describe a circuit to generate subthreshold gate voltages required for multiple sleep mode operation. Each mode applies different gate biases to the footer device, thereby resulting in different V_{GND} potentials and hence different leakage savings and wake-up overheads. We demonstrate the advantage of multiple sleep modes on several benchmark circuits. Given the growing demand of effective power management, the multiple sleep mode capability can be very useful in reducing overall power consumption of a chip.

The rest of the paper is organized as follows. In the next section, we explain the concept behind the proposed multiple power gating method. Section 3 describes the bias generator circuit required for the multiple sleep mode operation and Section 4 shows the final schematic. We present our results in Section 5 before concluding in Section 6.

2. The Concept

The proposed circuit generates multiple sleep modes by controlling the steady state virtual ground rail potential. The virtual ground potential, in turn, is controlled by the gate voltage of the footer device during sleep mode. In this section, we develop an analytical formulation that relates the virtual ground rail potential with the gate voltage of the footer device. The relationship of a sleep model virtual ground rail potential to corresponding leakage savings and wake-up overhead is also quantitatively analyzed.

This work was supported in part under Defense Advanced Research Project Agency Contract F33615-030C-4106.

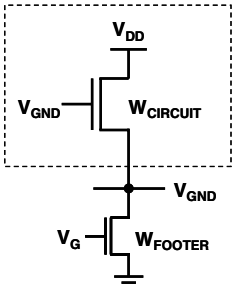


Figure 1: Simplified circuit for V_{GND} model

Let us assume that the footer device operates in the weak inversion region ($V_G < V_{TH}$) and the leakage of the logic circuit can be approximated by the leakage of a single transistor of effective width $W_{circuit}$ as shown in Figure 1. Under the assumption that the footer is biased in the weak inversion region, the steady state V_{GND} potential can be obtained by matching the leakage current of the logic circuit with the leakage of the footer device.

$$I_{leak}(Circuit) = I_{leak}(Footer) \tag{1}$$

Substituting $I_{leak} = I_0\left(\frac{W}{L}\right)10^{\frac{(V_{GS}-V_{TH})+\eta(V_{DS})}{S_S}}$ for the logic circuit and the footer in Equation 1 gives

$$I_0 \frac{W_{CIRCUIT}}{L} 10^{\frac{(-V_{THC})+\eta(V_{DD}-V_{GND})}{S_S}} = I_0 \frac{W_{FOOTER}}{L} 10^{\frac{(V_G-V_{THF})+\eta V_{GND}}{S_S}} \tag{2}$$

Here, V_{THC} and V_{THF} represent the threshold voltages of the logic circuit and the footer device respectively, η is the DIBL coefficient and S_S is the subthreshold slope. Solving Equation 2 for V_{GND} results in following expression:

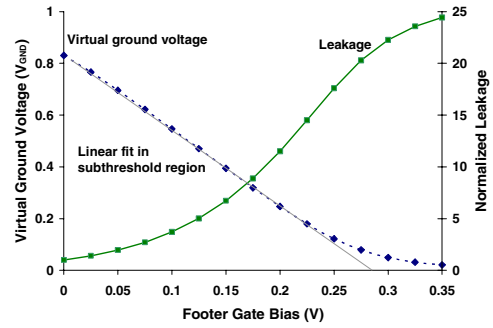
$$V_{GND} = \frac{-V_G + S_S \log_{10}\left(\frac{W_{CIRCUIT}}{W_{FOOTER}}\right) + (V_{THF} - V_{THC}) + \eta V_{DD}}{2\eta} \tag{3}$$

The above equation shows that the steady state V_{GND} is linearly dependent on footer gate voltage V_G with a negative slope. If the footer gate voltage is increased, it results in a decrease in the virtual ground potential and *vice versa*. Hence, V_{GND} potential in the sleep mode can be effectively controlled by the gate voltage of the footer device.

The ability to control V_{GND} potential of a sleep mode provides the capability to control the inherent trade-off between leakage savings and wake-up overhead. If we represent the leakage current of a circuit in the *active* mode by I_{active}, then the leakage savings in the sleep mode with a virtual ground rail potential V_{GND} is given by

$$\frac{I_{sleep}}{I_{active}} = 10^{-\left(\frac{\eta(V_{DD}-V_{GND})}{S_S}\right)} \tag{4}$$

The above equation shows that higher V_{GND} results in higher leakage savings. However, the wake-up time and wake-up energy for recovering from the V_{GND} sleep state is also higher. If the total capacitance of the circuit block (including parasitic capacitances) is represented by $C_{CIRCUIT}$, then wake-up time and wake-up energy can be expressed as



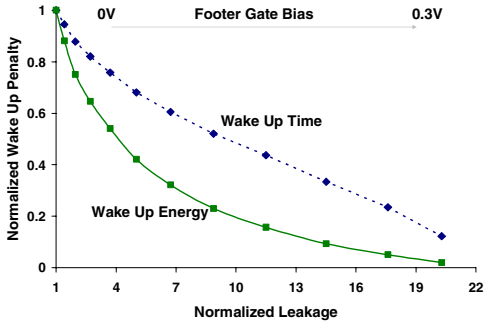


Figure 2: Effect of footer gate voltage on virtual ground potential (V_{GND}) and corresponding leakage vs wake-up penalty trade-off

$$T_{WakeUp} = \frac{C_{CIRCUIT} V_{GND}}{I_{ON,F}} \qquad E_{WakeUp} = \frac{1}{2} C_{CIRCUIT} V_{GND}^2 \tag{5}$$

Here, $I_{ON,F}$ represents the on-current of the footer device after the device has been turned-on to wake up the circuit.

To verify above claims, we consider a footed logic circuit in 65 nm technology with a V_{DD} of 1V. The logic block contains a 32-bit ripple carry adder. The size of the footer device was chosen to be ~12% of the total NMOS width in the adder block. Figure 2 shows the relationship between applied footer gate voltage and sleep mode steady state virtual ground rail potential. The figure shows that as footer gate bias is increased, the virtual ground rail potential decreases and the leakage power increases. A gate bias in excess of 0.3V causes the sleep mode steady state V_{GND} to discharge to nearly zero volts. The figure also shows the trade-off between leakage savings and wake-up overhead as obtained by varying gate voltage of the footer device in this range. It is clear from the figure that by appropriately biasing the footer in the weak inversion region, multiple sleep modes with different leakage savings and wake-up penalty trade-off can be easily generated.

3. The Bias Generator

In the previous section, we showed the importance of controlling footer gate voltage for generating multiple sleep modes. The implementation of this approach requires a robust gate-bias generator. In this section, we discuss the circuit for generating subthreshold gate voltages required for multiple sleep mode operation. Figure 3 shows the circuit diagram of such a robust gate-bias generator.

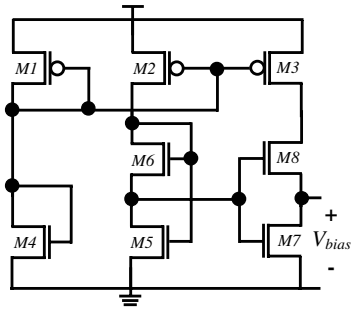


Figure 3: Robust bias generation circuit

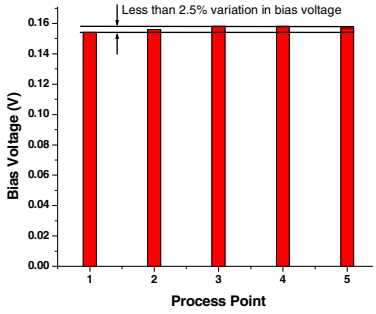


Figure 4: Robustness of bias generation circuit over the spread of process and temperature

Transistors *M1*, *M2* and *M3*, all equally sized, form a current mirror, biased by the V_{GS} of *M1*. The currents in each leg of this circuit then should nominally be the same. However, since transistors *M5* and *M6* as well as transistors *M7* and *M8* form stacked paths, and since transistors *M4* – *M7* are all equally sized while *M8* is larger, the currents through each leg of the current mirror are different. The output of the circuit is taken across device *M7*. Transistor *M8* can be sized appropriately depending on the necessary bias output needed. As noted previously, transistors *M1* through *M3* are equally sized and transistors *M4* through *M7* are equally sized, though different than the PMOS devices. Transistor *M8* is used to generate an appropriate V_{bias}. For example, we set $W_{M8} = 4 \times W_{M7}$ to generate a bias voltage of 155 mV.

We note that this bias voltage generation is fairly robust against six-sigma process variation as well as temperature. We simulated this bias circuit over five differing points in the process and temperature spread and found at most a 2.5% variation in the generated bias voltage. Figure 4 summarizes these results.

4. Multiple Sleep Modes

Once we have the capability to generate different gate voltages, multiple sleep modes can be easily obtained by applying a different bias to the footer in each mode. A simple block diagram is shown in Figure 5. This circuit has four operating modes – *Active, Sleep, Dream* and *Snore*. In each mode, different gate bias is applied to the gate terminal of the footer device. The intermediate gate voltages (V_1 and V_2, $V_1<V_2<V_{TH}$) for *Sleep* and *Dream* modes are generated using bias generator discussed in the previous section. A two-bit select signal is used to choose the desired operating mode.

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

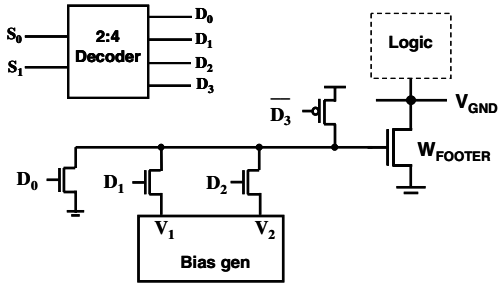


Figure 5: Multiple Sleep Mode Schematic

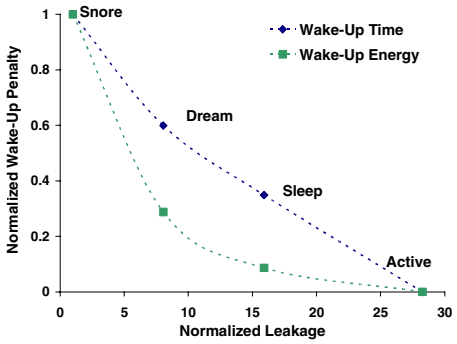


Figure 6: Leakage savings-wake-up characteristics of four operating modes

We implemented the proposed approach on a 32-bit ripple carry adder testcase discussed in Section 2. Figure 6 shows the leakage and wake up penalty for four different modes. It is clear from the figure that multiple sleep modes allow us to trade-off leakage savings with wake up latency and wake up energy overhead and hence can be very useful in wake-up constrained sleep operations. For this experiment, the values of two intermediate gate voltages (V_1 and V_2) were chosen to get two evenly spaced points in the wake-up penalty vs. leakage space. The *sleep* mode was state-retentive while the *dream* mode did not retain the state. However, the optimum number of intermediate modes and their leakage, wake-up and state-retention requirements depend on the application programs and should be chosen accordingly.

We tested the robustness of above circuit against process, voltage and temperature variations. We generated seven PVT corners by selecting various combinations of V_{DD} (0.9, 1.0 and 1.1V), temperature (55C, 85C and 115C) and process (weak, nominal and best) values. As expected, the absolute values of leakage and wake-up overhead showed a significant change with PVT variations. However, the proposed circuit is robust if it ensures that the relative trade-off between leakage and wake-up overhead in various sleep modes is maintained in the presence of these variations. Figure 7 shows the leakage and wake-up latency values for different modes at various PVT corners. Here, leakage and wake-up overhead numbers are normalized with

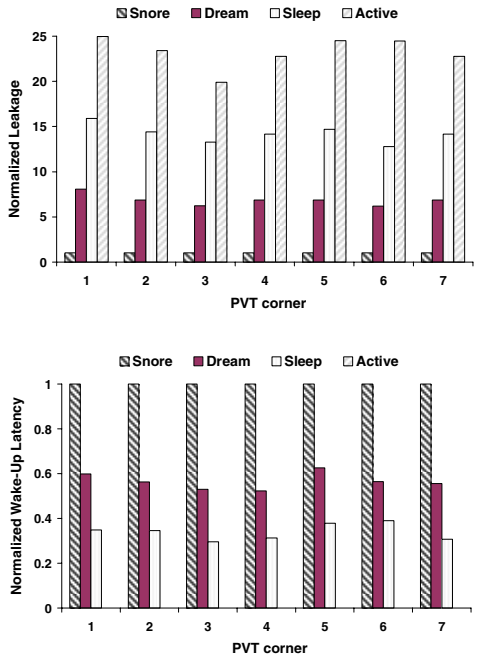


Figure 7: Leakage savings and wake-up characteristics of four operating modes at various PVT corners

respect to the corresponding values in the *snore* mode. There is no wake-up penalty in the active mode. It is clear from the figure that the multiple sleep mode circuit provides a desired trade-off between leakage savings and wake-up overhead at all operating conditions and process corners.

5. Benchmark Applications

One of the primary drawbacks of power gating is the long wake up latency associated with discharging of the virtual ground rail during mode change from *sleep* to *active*. This latency can be up to 8 – 10 cycles. This makes power gating effective only when the data input is not switching for greater than the latency time plus the path propagation delay, generally greater than 8 cycles.

To substantiate the importance of multiple sleep modes, we analyzed the data switching rate of a 64-bit ALPHA architecture processor executing various applications. We examined the data switching rate and applied hard (conventional) power gating when the data was constant for 8 or more cycles. Next, we took the same data and applied multiple sleep modes. The optimum sleep mode for the processor was selected based on the number of idle cycles and the wakeup requirements as described in Table I. The *snore* mode is same as regular power gating and requires 8 cycles to wake up. *Dream* takes a little over half the time to wake up and so it was allotted 5 cycles. Finally, the *sleep* mode was assigned 3 cycles to fully wake up. The number of cycles in the wake-up latency set the constraint on the minimum number of idle clock cycles needed before a processor can enter in the corresponding sleep mode without any wakeup overhead. For latencies of less than 3 cycles, no sleep mode is applicable.

Figure 8 shows the percent leakage savings in various sleep modes as a function of number of idle cycles. As the number of idle cycles grows, the average leakage savings increase for all modes because the one-time wake-up power overhead gets offset by the leakage savings in each idle cycle. Figure 8 is divided in four regions based on the wake-up latencies shown in Table I. In Region 1, no sleep mode can be used since the minimum wake-up latency (idle time) is 3 cycles. In Region 2, we use *sleep* mode even though the *dream* mode provides more leakage savings. This is because the *dream* mode requires at least 5 idle cycles to wake up, and thus can not be used in this Region. In Region 3, *snore* can not be used since it requires at least 8 idle cycles to wake up. In this region, we have a choice between *sleep* and *dream*, so we choose *dream* for the greater leakage savings. In Region 4, *snore* mode provides the best leakage savings.

We applied regular power gating and multiple mode power gating to six different applications. On average, we saw a reduction in leakage of about 17% using multiple mode power gating as compared to conventional single mode gating. Figure 9 shows the leakage savings for various benchmark applications. These additional power savings are obtained at no superfluous wake-up delay penalty because the sleep modes are selected under wake-up constraints.

Table I: Wake-up latency for various sleep modes

Mode	Wake-up Latency (cycles)
Active	-
Sleep	3
Dream	5
Snore	8

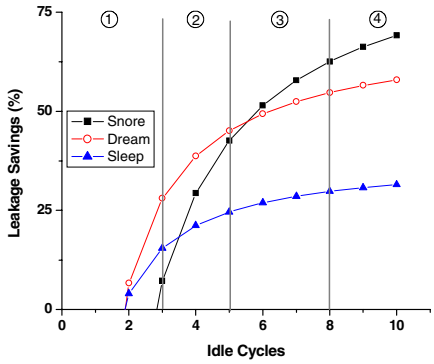


Figure 8: Percent leakage savings for various sleep modes as a function of number of idle cycles

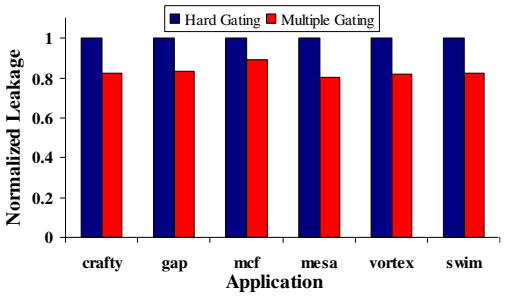


Figure 9: Leakage savings for various applications on a 64-bit Alpha processor using multiple off state power gating

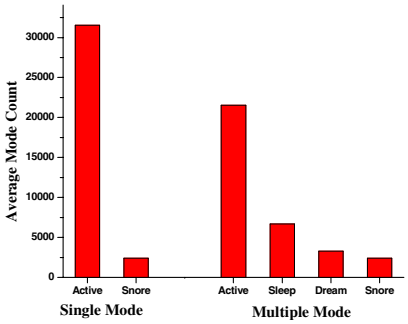


Figure 10: Average mode count across all six applications

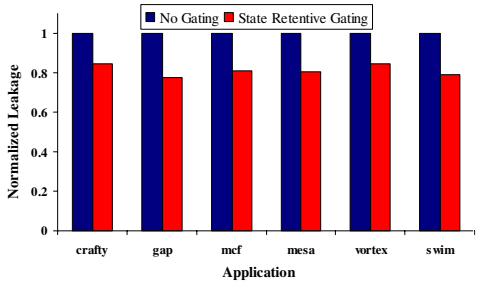


Figure 12: Leakage savings for various applications using a state retentive power gating mode

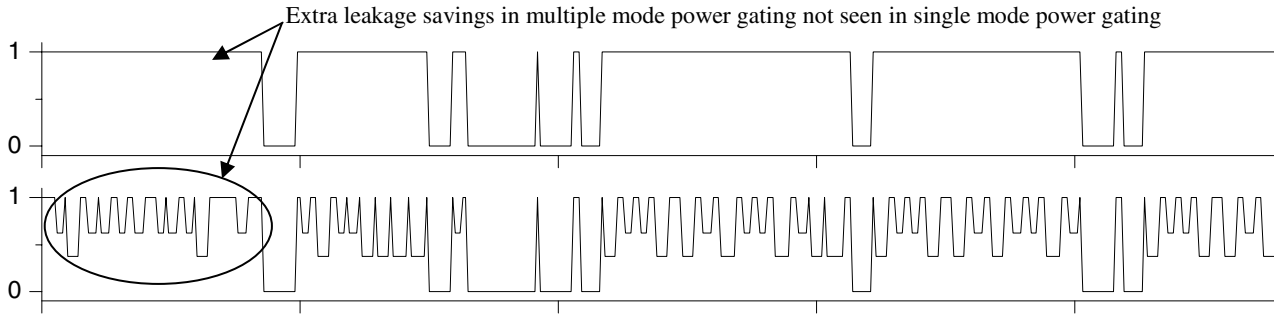


Figure 11: Sleep mode timeline of single (top) and multiple mode (bottom) power gating

We also looked at the total mode count for both single and multiple mode power gating. Over the six applications, single mode power gating was used an average of 2,414 times as compared to multiple mode power gating where a power savings mode (*sleep*, *dream* or *snore*) was selected an average of 12,393 times. These results show that the multiple mode power gating allows a processor to save additional leakage by entering a leakage saving state more often than the conventional gating. Figure 10 summarizes the average number of times a mode is utilized for both single mode and multi-mode power gating.

Multiple mode power gating allows a processor to enter a power gated mode more frequently than single mode power gating. Figure 11 demonstrates this result with a sleep mode timeline. In normal operation, the timeline has a value of 1. In full power gated mode, it has a value of 0. For the multiple mode power gating case, the timeline has a value in between 0 and 1, representing a partially power gated state.

Finally, some applications require the internal logic gates to retain state, even when there is no logical evaluation being performed. In such a case, single mode power gating cannot be used since that would flush out the data. However, our *sleep* mode is state retentive and can be used to retain state while simultaneously reducing leakage power consumption. For the six tested applications, we compared the leakage power of no power gating to state-retentive power gating (only *sleep* mode) and found an average of 19% reduction in leakage power while retaining states of the internal nodes. Figure 12 shows this data.

6. Conclusions

A multiple mode power gating design technique was introduced for enhanced leakage reduction. Our simulations and data traces show an average of 17% reduction in leakage power as compared to traditional single mode power gating. The multiple off mode capability also provides an option of state-retentive mode to enable power savings during inactive periods when the state of the circuit must be preserved. The flexibility provided by the multiple sleep modes can be very useful in effective power management in power conscious designs.

References

[1] S. Borkar, "Design challenges of technology scaling," *IEEE Micro*, Vol. 19, No. 4, pp. 23-29, 1999.

[2] K. Bernstein, C. T. Chuang, R. Joshi and R. Puri, "Design and CAD challenges in sub-90 nm CMOS technologies," *ICCAD,* pp 129-136, Nov. 2003.

[3] S. Mutoh *et al.*, "1-V power supply high-speed digital circuit technology with multithreshold-voltage CMOS," *JSSC*, vol. SC-30, pp. 847–854, Aug. 1995.

[4] J. Kao, S. Narendra, and A. Chandrakasan, "MTCMOS hierarchical sizing based on mutual exclusive discharge patterns," *DAC*, pp. 495–500, June 1998.

[5] H. Kawaguchi, K. Nose, and T. Sakura, "A super cut-off CMOS (SCCMOS) scheme for 0.5-V supply voltage with picoampere stand-by current," *JSSC*, vol. SC-35, pp. 1498–1501, Oct. 2000.

[6] S. Kosonocky, M. Immediato, P. Cottrell, T. Hook, R. Mann, and J. Brown, "Enhanced multi-threshold (MTCMOS) circuits using variable well bias," *ISLPED*, pp. 165–169, Aug. 2001.

[7] M. Anis, S. Areibi, M. Mahmoud, and M. Elmasry, "Dynamic and leakage power reduction in MTCMOS circuits using an automated efficient gate clustering technique," *DAC*, pp. 480–485, June 2002.

[8] S. Kim, S. Kosonocky, D. Knebel, and K. Stawiasz, "Experimental measurement of a novel power gating structure with intermediate power saving mode", *ISLPED*, pp. 20-25, August 2004.