Sleepy Stack Reduction of Leakage Power

Jun Cheol Park, Vincent J. Mooney III, and Philipp Pfeiffenberger

Center for Research on Embedded Systems and Technology
School of Electrical and Computer Engineering, Georgia Institute of Technology
Atlanta, GA 30332-0250 USA
{jcpark,mooney}@ece.gatech.edu, gte125y@mail.gatech.edu

Abstract. Leakage power consumption of current CMOS technology is already a great challenge. ITRS projects that leakage power consumption may come to dominate total chip power consumption as the technology feature size shrinks. We propose a novel leakage reduction technique, named “sleepy stack,” which can be applied to general logic design. Our sleepy stack approach retains exact logic state – making it better than traditional sleep and zigzag techniques – while saving leakage power consumption. Unlike the stack approach (which saves state), the sleepy stack approach can work well with dual-V_{th} technologies, reducing leakage by several orders of magnitude over the stack approach in single-V_{th} technology. Unfortunately, the sleepy stack approach does have a area penalty (roughly 50∼120%) as compared to stack technology; nonetheless, the sleepy stack approach occupies a niche where state-saving and extra low leakage is desired at a (potentially small) cost in terms of increased delay and area.

1 Introduction

The advent of a mobile computing era has become a major motivation for low power design because the operation time of a mobile device is heavily restricted by its battery life. The growing complexity of mobile devices, such as a cell phone with a digital camera or a personal digital assistant (PDA) with global positioning system (GPS), makes the power problem more challenging.

Dynamic power consumption was previously a major concern for chip designers since dynamic power accounted for 99% or more of the total chip power. However, as the feature size shrinks, static power, which consists mainly of subthreshold and gate-oxide leakage power, has become a great challenge for current and future technologies. The main reason is that leakage current increases exponentially as the feature size shrinks. Based on the International Technology Roadmap for Semiconductors (ITRS), Kim et al. report that subthreshold leakage power dissipation of a chip will exceed dynamic power dissipation at the 65nm feature size [1][2].

Techniques for leakage power reduction can be grouped in two categories: state-preserving techniques where circuit state (present value) is retained and state-destructive techniques where the current boolean output value of the circuit might be lost [1]. A state-preserving technique has an advantage over a state-destructive technique in that with a state-preserving technique the circuitry can resume operation at a point much later in time without having to somehow regenerate state. Our new design technique, which we call the "sleepy stack" technique, retains data during sleep mode while providing reduced leakage power consumption at a cost of slightly increased delay. Furthermore, the sleepy stack approach can be applicable to single- and dual-threshold voltage technologies. The sleepy stack approach delivers a new choice to designers to implement low-leakage-power circuits that retains state.

The rest of the paper is organized as follows: Section 2 discusses sources of leakage power and previous low-leakage approaches. Section 3 describes the proposed sleepy stack approach with comparisons to previous approaches. Section 4 presents the simulation methodology, and Section 5 explains the results. Section 6 concludes the paper.

2 Previous Work

In today's technology, the main contributer to static power consumption of a CMOS circuit is subthreshold leakage, i.e., source to drain current when gate voltage is smaller than the transistor threshold voltage. The subthreshold leakage current can be expressed as follows:

$$I_{sub} = K_1 W e^{-V_{th}/nV_\theta}(1 - e^{-V/N_\theta}) \tag{1}$$

where K_1 and n are experimental values, W is the width of the transistor, V_{th} is the threshold voltage and V_θ is the thermal voltage [1]. Since subthreshold current increases exponentially as the threshold voltage decreases, deep sub-micron technologies with scaled down threshold voltages will severely suffer from subthreshold leakage power consumption. In addition to subthreshold leakage, another contributer to leakage power is gate-oxide leakage power due to the tunneling current through the gate-oxide insulator. Since gate-oxide thickness will be reduced as the technology decreases, deep sub-micron technology will also suffer from gate-oxide leakage power. However, previously proposed work for leakage power reduction at the circuit level has focused on subthreshold leakage power because gate-oxide leakage is relatively small compared to subthreshold leakage [1]. Although gate-oxide leakage will also increase exponentially as technology feature size decreases, a solution for gate-oxide leakage may lie in the development of high dielectric constant (high-k) gate insulators. In this paper, we focus exclusively on reducing subthreshold leakage power.

As introduced in Section 1, previously proposed work can be divided into techniques that either (i) preserve state or (ii) destroy state. State-destructive techniques (ii) cut off transistor (pull-up or pull-down or both) networks from supply voltage or ground using sleep transistors [5]. These types of techniques are also called gated-V_{DD} (note that a gated clock is generally used for dynamic power reduction). Motoh et al. propose a technique called multi-threshold-voltage CMOS (MTCMOS), which adds high V_{th} sleep transistors between pull-up networks and V_{DD} and between pull-down networks and ground while logic circuits use low V_{th} to maintain logic performance [5]. The sleep transistor is turned off when the logic circuits are not in use. Powell et al. propose a gated-V_{DD} cache technique called DRI cache, which dynamically changes cache size with gated-V_{DD} transistors of dual or single V_{th} [4]. Since the gated-V_{DD} technique cuts off logic blocks from V_{DD} and Gnd, time and energy for waking up are significant. The zigzag technique proposes the reduction of wake-up overhead by choosing a particular circuit state (e.g., corresponding to a "reset") and then, for the exact circuit state chosen, cutting off the pull-down network for each gate whose output is high while conversely cutting off the pull-up network for each gate whose output is low [9]. Although the zigzag technique can retain a particular state chosen prior to chip fabrication, any other arbitrary state during regular operation is lost in power-down mode. Another technique to reduce leakage power is transistor stacking that exploits the stack effect, i.e., it turns out that two stacked and turned off transistors reduce subthreshold leakage current substantially due mainly to a reverse bias between the gate and source [6]. Narendra et al. study the effectiveness of the stack effect including effects from increasing the channel length [7]. Since forced stacking of what previously was a single transistor increases delay, Johnson et al. propose an algorithm that finds circuit input vectors that maximize stacked transistors of existing complex logic [8]. Flautner et al. propose the "drowsy cache" technique that switches supply voltage instead of gating V_{DD} [10]. The effect of drowsy caches in terms of leakage power reduction is smaller than gated-V_{DD} techniques, but the drowsy cache technique can retain the original state thus can be used for designing memories.

3 Sleepy Stack

In this section, our new low-leakage-power design, named "sleepy stack," is described and compared with well-known previous approaches, i.e., the sleep, zigzag and stack techniques explained in Section 2. The base case, shown in Fig. 1, consists of three stages each with a pull-up network and a pull-down network and with the final stage connected to a load capacitance. Fig. 2, 3 and 4 show previous low-leakage approaches applied to the base case. The sleep approach in Fig. 2

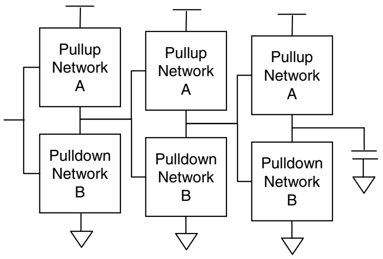

Fig. 1. Base case

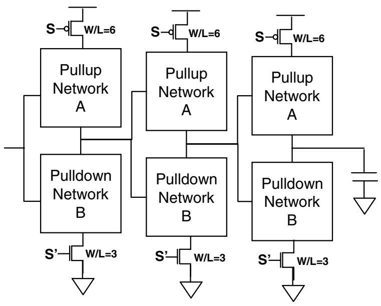

Fig. 2. Sleep

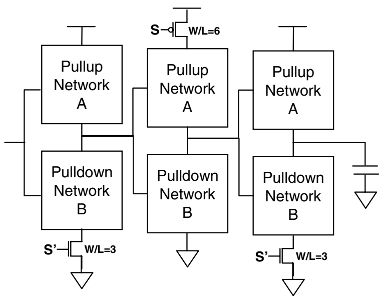

Fig. 3. Zigzag

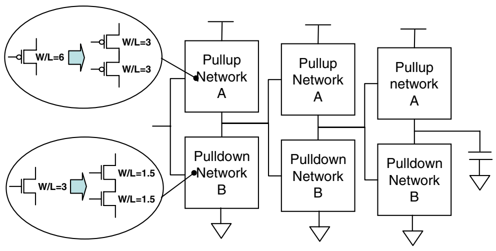

Fig. 4. Stack

uses sleep transistors between both V_{DD} and the pull-up network as well as between Gnd and the pull-down network. Generally, the width/length (W/L) ratio is sized based on a trade-off between area, leakage reduction and delay. During sleep mode, sleep transistors are turned off and leakage current is suppressed. However, the additional sleep transistors increase area and delay. Furthermore, the pull-up and pull-down network will have floating values and lose state during sleep mode. The zigzag approach in Fig. 3 first analyzes each gate for a particular input assumed during sleep mode and either assigns a sleep transistor to the pull-down network if the output is "1" or else assigns a sleep transistor to the pull-up network if the output is "0." In Fig. 3, we assume that, in the sleep mode, the input of the logic is "0" and each logic stage reverses its input signal, i.e., the output is "1" if the input is "0," and the output is "0" is the input is "1." Accordingly, the pull-down network of the first stage is off, and so a sleep transistor is added. Thus, the zigzag approach uses fewer sleep transistors than the original sleep approach. However, analyzing logical state and finding input vectors of complex logic blocks are NP-hard problems which need to be solved in order to apply the zigzag approach [8].

Fig. 4 shows the stack approach, which forces a stack effect by breaking down an existing transistor into two half size transistors. The forced stack approach can achieve huge leakage power saving while retaining the logic state. However, the divided transistors increase delay significantly and may restrict the usefulness of the forced stack approach.

The key idea of the sleepy stack technique is to combine the sleep transistor approach during active mode with the stack approach during sleep mode. The structure of the sleepy stack approach is shown in Fig. 5. The sleepy stack technique divides existing transistors into two transistors each typically with the same width W_1 half the size of the original single transistor's width W_2 (i.e., $W_1 = W_2/2$). Then sleep transistors are added in parallel to one of the transistors in each set of two stacked transistors; see Fig. 6 for an example. The divided transistors reduce leakage power using the stack effect while retaining state. The added sleep transistors operate similar to the sleep transistors used in the sleep technique in which sleep transistors are turned on during active mode and turned off during sleep mode. Fig. 6 depicts the sleepy stack operation

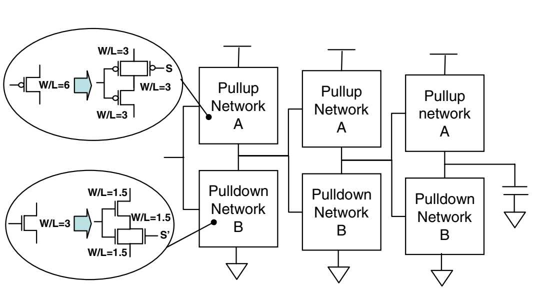

Fig. 5. Sleepy stack

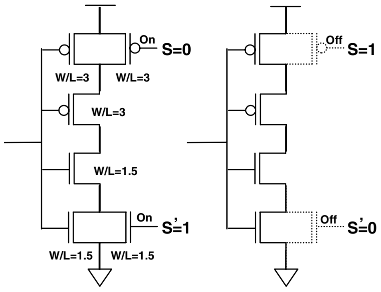

Fig. 6. Sleepy stack active mode (left) and sleep mode (right)

using an inverter. During active mode, S=0 and S'=1 are asserted, and thus all sleep transistors are turned on. Due to the added sleep transistor, the resistance through the activated (i.e., "on") path decreases, and the propagation delay decreases (compared to not adding sleep transistors while leaving the rest of the circuitry the same, i.e., with stacked transistors). During the sleep mode, S=1 and S'=0 are asserted, and so both of the sleep transistors are turned off. The stacked transistors in the sleepy stack approach suppress leakage current.

4 Experimental Methodology

We compare the proposed sleepy stack approach to a base case (based on Fig. 1) and three of the previous approaches explained earlier, namely zigzag, sleep and stack. Thus, we compare five design approaches in terms of power consumption (dynamic and static), delay and area.

To show that the sleepy stack approach is applicable to general logic design, we choose three generic circuits: (i) a chain of 3 inverters, (ii) a 4-input multiplexer and (iii) a 4-bit adder. The logic diagram of the 4-input multiplexer used for (ii) is shown in Fig. 7. One bit of the 4-bit adder used for (iii) is shown in Fig. 8 of which is sized as noted in the same figure.

The inverter chain uses three inverters each with W/L=6 for PMOS and W/L=3 for NMOS for the base case. Sleep transistors in the sleep approach (Fig. 2) and the zigzag approach (Fig. 3) are sized such that any sleep transistor between V_{DD} and a pull-up network takes the size of the largest transistor in the pull-up network, and any sleep transistor between Gnd and a pull-down network takes the size of the largest transistor in the pull-down network. For example, sleep transistors used in the pull-up and pull-down networks of the base case inverter chain have W/L=6 and W/L=3, respectively as shown in Fig. 2. Transistors in the stack approach are sized to half of the size of the base case transistors, e.g., transistors used in pull-up and pull-down of the base case inverter chain have W/L=3 and W/L=1.5, respectively, as shown in Fig 4. Similarly, transistors, including sleep transistors, in the sleepy stack approach are sized to half of the size of the base case transistors as shown in Fig. 5. The

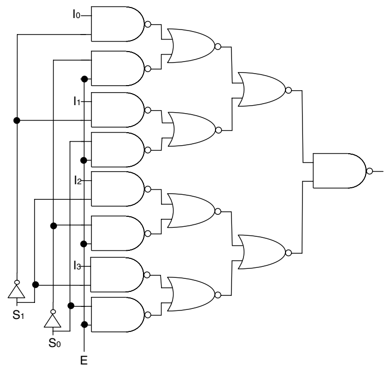

Fig. 7. 4-input multiplexer

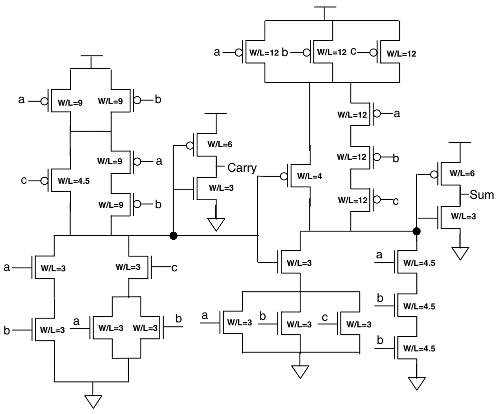

Fig. 8. 1-bit full adder

load capacitance is set to $3C_{inv}$ (where C_{inv} is the capacitance of an inverter with pull-up W/L=3, pull-down W/L=1.5).

To estimate area, delay and power, we first design target benchmark circuits using Cadence Virtuoso, a custom layout tool [12], and North Carolina State University Cadence design kit targeting TSMC 0.18μ technology [13]. Then we extract schematics from layout to obtain transistor circuit netlists. we use HSPICE simulation to estimate delay and power of the benchmarks, for which we use Avant! Stat-HSPICE [11].

Since we don't have access to the layout design proprietary below 0.18μ technology, we use the Berkeley Predictive Technology Model (BPTM) parameters to estimate delay and power for technologies below 0.18μ [14,15]. We choose four different technologies from BPTM to observe changes of power and delay as technology shrinks. The chosen technologies, i.e., 0.07μ, 0.10μ, 0.13μ and 0.18μ, use supply voltages of 1.0V, 1.3V, 1.6V and 1.8V, respectively. We assume that only a single supply voltage is available in each technology. We do consider both single- and dual-V_{th} technology for the zigzag, sleep and sleepy stack approaches (note that for the straightforward stack approach, no transistors exist which could be made high-V_{th} without a dramatic increase in delay). With dual-V_{th} technology, the zigzag and sleep approaches use high-V_{th} sleep transistors. Similarly, in the sleepy stack approach with dual-V_{th} technology, high-V_{th} transistors are used for sleep transistors and transistors that are parallel to the sleep transistors. We set all high-V_{th} transistors to have 2.5 times higher V_{th} than the V_{th} of a normal transistor just like [5]. More details about the experimental methodology can be found in a technical report [16].

5 Experimental Results

We measure worst case propagation delay and power for the five design approaches, which are the base case, zigzag, sleep, stack and sleepy stack ap-

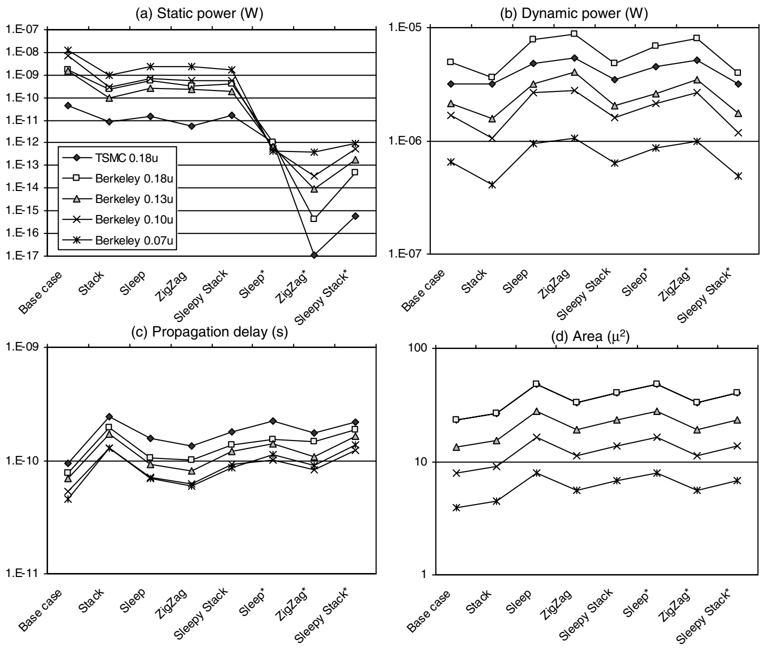

Fig. 9. Results of chain of 3 inverters (*dual V_{th})

proaches. Dynamic power is measured with a random input vector changing every clock cycle, i.e., 4ns; static power is measured using a sampling of input vectors and averaged. The area of the benchmarks below 0.18μ technology is estimated by scaling area of benchmark layout using TSMC 0.18μ technology. We add 10% of area overhead considering non-linear scaling layers, e.g., a metal layer.

Fig. 9, 10 and 11 show the experimental results. Focusing on the single V_{th} 0.07μ technology implementation of each benchmark shown in Table 1, we see that our sleepy stack approach results in leakage power roughly equivalent to the other three leakage-reduction approaches in the same technology. Compared to the sleep and zigzag approaches, which do not save state, the sleepy stack approach results in up to 67% delay increase and up to 69% area increase. Thus, we do not recommend the sleepy stack approach with single-V_{th} when state-preservation is not needed. Compared to the stack approach, which saves state, the sleepy stack approach results in up to 120% area increase, but the sleepy stack is up to 32% faster. Note that if we increase stack widths (thus decreasing transistor resistances), we obtain the results shown in Table 2. In this case, rather than improve, the stack approach shows increased leakage with a slight improvement in delay!

As explained in Section 4, the zigzag, sleep and sleepy stack approaches are also implemented using dual-V_{th} technology. The main advantage of the sleepy stack approach over the stack approach is that dual-V_{th} technology can be effec-

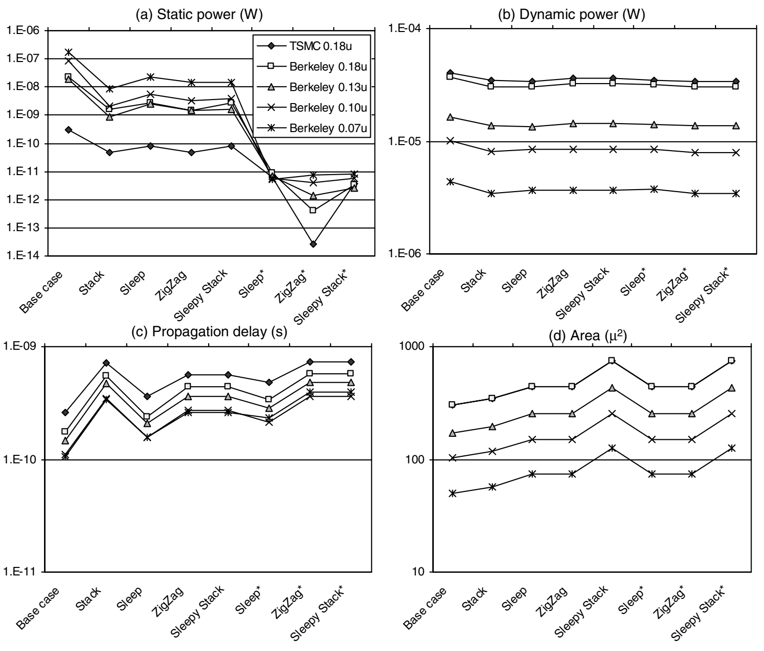

Fig. 10. Results of chain of 4-input multiplexers (*dual V_{th})

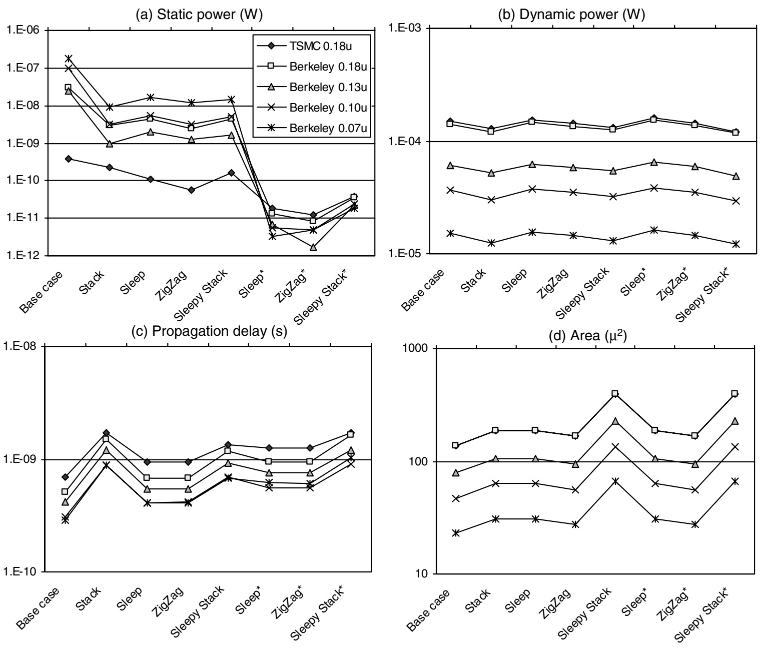

Fig. 11. Results of chain of 4-bit adders (*dual V_{th})

Table 1. Area, delay and power estimation (0.07μ)

A chain of 3 inverters	Propagation delay (s)	Static Power (W)	Dynamic Power (W)	Area (µ2)
Base case	4.61E-11	1.24E-08	6.56E-07	3.92
Stack	1.28E-10	9.89E-10	4.08E-07	4.48
Sleep	6.98E-11	2.40E-09	9.49E-07	8.00
ZigZag	5.99E-11	2.27E-09	1.05E-06	5.54
Sleepy Stack	8.75E-11	1.77E-09	6.35E-07	6.78
Sleep (dual Vth)	1.14E-10	4.32E-13	8.58E-07	8.00
ZigZag (dual Vth)	9.03E-11	3.84E-13	9.87E-07	5.54
Sleepy Stack (dual Vth)	1.38E-10	9.88E-13	4.88E-07	6.78

4-input multiplexer	Propagation delay (s)	Static Power (W)	Dynamic Power (W)	Area (µ2)
Base case	1.05E-10	1.72E-07	4.35E-06	50.17
Stack	3.39E-10	8.63E-09	3.43E-06	57.40
Sleep	1.56E-10	2.24E-08	3.66E-06	74.11
ZigZag	2.58E-10	1.41E-08	3.64E-06	74.36
Sleepy Stack	2.58E-10	1.51E-08	3.64E-06	125.33
Sleep (dual Vth)	2.35E-10	5.03E-12	3.73E-06	74.11
ZigZag (dual Vth)	3.97E-10	7.54E-12	3.43E-06	74.36
Sleepy Stack (dual Vth)	3.97E-10	8.19E-12	3.43E-06	125.33

4-bit adder	Propagation delay (s)	Static Power (W)	Dynamic Power (W)	Area (µ2)
Base case	2.91E-10	1.81E-07	1.52E-05	22.96
Stack	8.89E-10	9.25E-09	1.24E-05	30.94
Sleep	4.11E-10	1.69E-08	1.54E-05	30.94
ZigZag	4.06E-10	1.20E-08	1.47E-05	27.62
Sleepy Stack	6.79E-10	1.50E-08	1.31E-05	65.88
Sleep (dual Vth)	6.20E-10	3.31E-12	1.61E-05	30.94
ZigZag (dual Vth)	6.15E-10	4.92E-12	1.47E-05	27.62
Sleepy Stack (dual Vth)	1.03E-09	1.88E-11	1.22E-05	65.88

Table 2. Area, delay and leakage power with various stack width (a chain of 3 inverters, 0.07μ)

	Base case	Stack (1X)	Stack (2X)	Stack (3X)	Stack (4X)	Sleepy stack (single Vth)	Sleepy stack (dual Vth)
Area (μ^2)	3.92	4.48	5.37	6.27	7.17	6.78	6.78
Delay (S)	4.61E-11	1.28E-10	1.14E-10	1.10E-10	1.07E-10	8.75E-11	1.38E-10
Leakage (W)	1.24E-08	9.89E-10	1.98E-09	2.97E-09	3.96E-09	1.77E-09	9.88E-13

tively applied to the sleepy stack, resulting in three orders of magnitude reduction in leakage when compared to the stack approach as seen in Figs. 9(a), 10(a) and 11(a) with small (7∼17%) associated increases in delay. Not surprisingly, the sleepy stack approach has 50∼120% larger area as compared to the stack approach. Therefore, our sleepy stack approach with dual-V_{th} can be used where state-preservation and ultra-low leakage power consumption are needed and are judged to be worth the area overhead.

One observation we notice from the results is that none of the approaches shows the best result in all criteria. Designers need to choose an appropriate technique for a given technology and a particular chip. Our sleepy stack approach provides a new low-power VLSI design technique to achieve significant leakage power reduction in deep sub-micron while achieving either (i) saving of state (unlike sleep and zigzag) or (ii) lower delay than a straightforward stack approach.

6 Conclusions

In nanometer scale CMOS technology, subthreshold leakage power consumption is a great challenge. Although previous approaches are effective in some ways, no perfect solution for reducing leakage power consumption is yet known. Therefore, designers choose techniques base upon technology and design criteria. Our novel sleepy stack, which combines the sleep and the stack approaches, is proposed as a new choice for logic designers. Furthermore, the sleepy stack is applicable to single and multiple threshold voltages. In conclusion, the sleepy stack combine some of the advantages of sleep transistors – most notably the effective use of dual-V_{th} technology – with some of the advantages of the stack approach – most notably the ability to save state. As such, the sleepy stack approach represents a new weapon in the VLSI designer's repertoire.

References

1. N. S. Kim et al., "Leakage Current: Moore's Law Meets Static Power," *IEEE Computer*, Vol. 36, Issue 12, pp. 68-75, December 2003.
2. International Technology Roadmap for Semiconductors by Semiconductor Industry Association, http://public.itrs.net, 2002.
3. L. T. Clark et al., "An Embedded 32-b Microprocessor Core for Low-Power and High-Performance Applications," *IEEE Journal of Solid-State Circuits*, Vol. 36, No. 11, pp. 1599-1608, November 2001.
4. M. Powell, S.-H. Yang, B. Falsafi, K. Roy and T. N. Vijaykumar, "Gated-Vdd: A Circuit Technique to Reduce Leakage in Deep-submicron Cache Memories," *International Symposium on Low Power Electronics and Design*, pp. 90-95, July 2000.
5. S. Mutoh et al., "1-V Power Supply High-speed Digital Circuit Technology with Multithreshold-Voltage CMOS," *IEEE Journal of Solis-State Circuits*, Vol. 30, No. 8, pp. 847-854, August 1995.
6. Z. Chen, M, Johnson, L. Wei and K. Roy, "Estimation of Standby Leakage Power in CMOS Circuits Considering Accurate Modeling of Transistor Stacks," *International Symposium on Low Power Electronics and Design*, pp. 239-244, 1998.
7. S. Narendra, S. Borkar, V. De, D. Antoniadis and A. Chandrakasan, "Scaling of Stack Effect and its Application for Leakage Reduction," *International Symposium on Low Power Electronics and Design*, pp. 195-200, August 2001.
8. M. Johnson, D. Somasekhar, L-Y. Chiou and K. Roy, "Leakage Control with Efficient Use of Transistor Stacks in Single Threshold CMOS," *IEEE Transactions on VLSI Systems*, Vol. 10, No. 1, pp. 1-5, February 2002.
9. K.-S. Min, H. Kawaguchi and T. Sakurai, "Zigzag Super Cut-off CMOS (ZSC-CMOS) Block Activation with Self-Adaptive Voltage Level Controller: An Alternative to Clock-gating Scheme in Leakage Dominant Era," *IEEE International Solid-State Circuits Conference*, Vol. 1, pp. 400-401, February 2003.
10. K. Flautner, N. S. Kim, S. Martin, D. Blaauw and T. Mudge, "Drowsy Caches: Simple Techniques for Reducing Leakage Power," *International Symposium on Computer Architecture*, pp. 148-157, May 2002.
11. Avant! Corporation, http://www.avanticorp.com.
12. Cadence Design Systems, http://www.cadence.com.

13. NC State University Cadence Tool Information, http://www.cadence.ncsu.edu.
14. Berkeley Predictive Technology Model (BPTM), http://www-device.eecs.berkeley.edu/˜ptm/.
15. Y. Cao, T. Sato, D. Sylvester, M. Orshansky, and C. Hu, "New paradigm of predictive MOSFET and interconnect modeling for early circuit design," *Proc. of IEEE CICC*, pp. 201-204, June 2000.
16. P. Pfeiffenberger, J. Park and V. Mooney, "Some Layouts Using the Sleepy Stack Approach," Technical Report GIT-CC-04-05, Georgia Institute of Technology, June 2004. http://www.cc.gatech.edu/tech_reports/index.04.html