Lookup Table-Based Adaptive Body Biasing of Multiple Macros

Byunghee Choi and Youngsoo Shin
Department of Electrical Engineering, KAIST
Daejeon 305-701, Korea

Abstract— **A reduced supply voltage must be accompanied by a reduced threshold voltage, which makes this approach to power saving susceptible to process variation in transistor parameters, as well as resulting in increased subthreshold leakage. We propose a new adaptive body biasing scheme, based on a lookup table for independent control of multiple functional blocks on a chip, which controls leakage and also compensates for process variation at the block level. An adaptive body bias is applied to blocks in active mode and a large reverse body bias is applied to blocks in standby mode. This is achieved by a central body bias controller, which has a low overhead in terms of area, delay, and power consumption. The problem of optimizing the required set of bias voltages is formulated and solved. A design methodology for semicustom design using standard-cell elements is developed and verified with benchmark circuits.**

I. Introduction

The supply voltage of CMOS circuits keeps being reduced in step with technology scaling so as to manage their power consumption. This increases the circuit delay, and the threshold voltage is reduced to compensate. This leads to an exponential increase in subthreshold leakage, which is the main component of standby power consumption. A reduced supply voltage has another implication in the design of circuits: process variations due to transistor parameters such as channel length and threshold voltage have a higher impact on speed and leakage current [1]. The spread in frequency and leakage distribution due to process variation can cause a 20× variation in chip leakage and a 30% variation in chip frequency [2]. This wide variation in frequency and leakage affects the yield, since chips with excessive leakage and chips at too low a frequency have to be discarded.

In order to accommodate the process variation and/or to reduce the leakage current, body bias circuits are used to control body (or substrate) bias dynamically [3]. The threshold voltage of an MOS transistor is a function of its body to source potential. The threshold voltage can be modulated to achieve higher performance by a forward body bias (FBB). The switching power can be reduced by means of FBB, since it allows the same frequency to be achieved at a lower supply voltage [4]. A reverse body bias (RBB) uses a higher threshold voltage and further reduces standby leakage current: the leakage current of a circuit is monitored and a feedback controller adjusts the body voltage until the predetermined leakage target is met [5]. It is possible to utilize FBB and RBB together, and this is called adaptive body bias (ABB), which has been shown to be very effective for minimizing the impact of both die-to-die and within-die parameter variations on frequency and active leakage power [6].

Although body biasing is efficient, the biasing circuits represent a large overhead in terms of area, power consumption, and the delay required to adjust the body bias. Thus, most circuit techniques for body biasing are targeted to an entire chip or several functional blocks, where the overhead of the biasing circuits is acceptable because of the scale of the circuits that they control, but the downside is that blocks are not controlled independently. In order to achieve fine-grain control of leakage and to compensate for intra-die process variation, it is important to be able to control several functional blocks on the same chip independently, which is only possible if biasing circuits with very low overheads can be used.

In this paper, we propose a new ABB scheme in which multiple macros are controlled independently, depending on their mode of operation. ABB is used to compensate for the process variation in the performance of a macro when it is in active mode and RBB is used to reduce its leakage current in standby mode. The salient feature of the proposed scheme is a lookup table that holds a binary code for each macro corresponding to its active mode body bias voltage. The binary code is fetched by a power management unit, and then the corresponding body bias voltage is generated by the controller. The code length and the set of bias voltages must be designed carefully to ensure maximum process compensation while keeping the overhead of bias controller tolerable; we show how this problem can be formulated and solved. We also propose a design methodology for applying our scheme to designs based on standard-cell elements.

The remainder of this paper is organized as follows: in the next section we describe our lookup table-based adaptive body bias scheme, and cover the overall operation of the circuit, the body bias generator, and various issues that arise in implementing the proposed scheme using standard-cell elements. In Section III, the optimization of body bias voltages is discussed. Experimental results are presented in Section IV, and we draw conclusions in Section V.

II. Lookup Table-Based Adaptive Body Biasing

A. Overall Operation

Fig. 1 outlines the way in which a lookup table can be used for adaptive body biasing. Suppose we have n independent macro functional blocks (macros for brevity) on a chip. A

Proceedings of the 8th International Symposium on Quality Electronic Design (ISQED'07)
0-7695-2795-7/07 $20.00 © 2007 IEEE

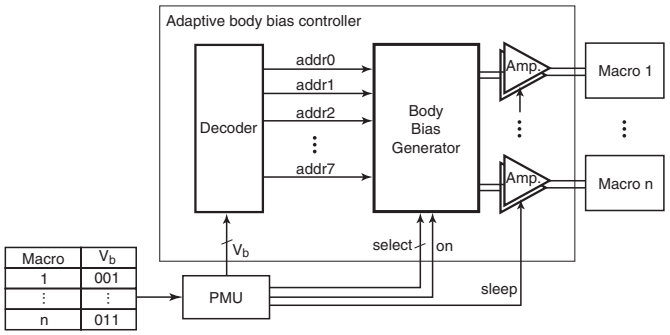

Fig. 1. Adaptive body biasing using a lookup table.

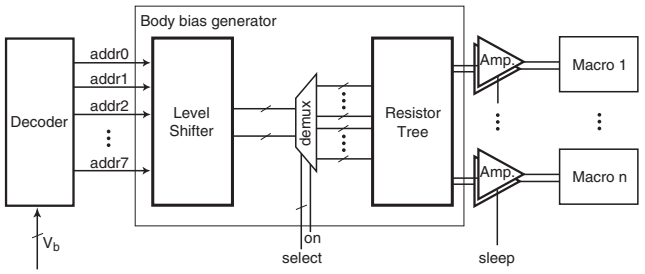

Fig. 2. Body bias generator.

power management unit (PMU) detects[1] a state change of a macro. When a macro changes its state from standby to active mode, the PMU fetches a codeword from the lookup table. The codeword is input to the adaptive body bias controller, which is marked as a block in Fig. 1. The controller then generates a pair of active-mode body bias voltages for the macro (one for NMOS and the other for PMOS transistors). When a macro changes its state from active to standby mode, a predetermined large reverse body bias is directly generated by the controller without using the lookup table.

The lookup table holds a codeword for each macro corresponding to the active mode body bias of that macro. The number of bits in each codeword determines the number of available bias voltages for compensating for process variations. Obviously, more bias voltages allow finer compensation for compensating process variation, but more bits means a larger lookup table and a larger overhead for the adaptive body bias controller. Thus, the length of the codeword needs to be determined carefully; this topic will be discussed in more detail in Section III. The values of the lookup table entries are determined and programmed after fabrication. For example, the delay of each macro is monitored for each codeword, and the code that allows the macro to meet its delay target is selected.

The proposed architecture allows multiple macros, each of which operates in more than one modes, to be controlled independently. In active mode, either FBB or RBB is used for process compensation, depending on the process variation of the macro. In standby mode, a large RBB is used to suppress the leakage current.

B. Body Bias Controller

Once the PMU has fetched a codeword for a macro, the decoder shown in Fig. 1 generates an *address* which has one bit at logic '1' for each combination of values in the codeword. This address is then used by the body bias generator to generate the body biases.

Fig. 2 shows the body bias generator in detail. It consists of a level shifter, a demultiplexer (DEMUX), and a resistor tree. The resistor tree requires voltages of V_{DDH} (higher than V_{DD}) and V_{DDL} (lower than V_{SS}), instead of V_{DD} and V_{SS}. A level shifter is employed to convert the address from the decoder, which uses V_{DD} as logic '1' and V_{SS} as logic '0', to a new *pair of addresses*: one for the PMOS switches in the resistor tree and the other for the NMOS switches. The address for the PMOS switches uses the levels V_{DDH} and V_{SS}; the address for the NMOS switches uses V_{DD} and V_{DDL}. The details will be explained in II-B.1.

After generation, the addresses are routed to the resistor tree through the DEMUX. Note that the resistor tree requires a pair of addresses for each macro, and so there are $2n$ addresses between the DEMUX and the resistor tree.

The `select` signal, which is $\lceil log_2 n \rceil$ bits wide, selects the macro to which level-shifted addresses are routed. The `on` signal, which turns on the DEMUX, is important in the operation of the body bias generator. Normally the DEMUX is turned off by de-asserting the `on` signal, decoupling the resistor tree from the level shifter. When the PMU wants to apply the active body bias to a particular macro, the corresponding values appear on the `select` lines. However, it takes time for the decoder and the level shifter to generate the required signals. Thus, the `on` signal must only be asserted after the delay for decoding and level shifting, so that the selected macro receives the correctly decoded and level-shifted addresses. Once the DEMUX has transferred the required addresses, `on` is de-asserted again, turning off the DEMUX.

1) Resistor Tree: In order to generate the active-mode body bias voltage, we use a resistor tree, as shown in Fig. 3. This tree consists of N equal transistors connected in series, which divide the potential difference between V_{DDH} and V_{DDL} into N intermediate potentials. A set of predetermined bias voltages can then be obtained by connecting switches where needed. The choice of bias voltages and N, which determines the resolution of bias voltages, will be discussed in the next section.

We use a PMOS switch to obtain the PMOS body bias voltage V_p, since the bias voltage for the PMOS body is around V_{DD}, although it will be higher than V_{DD} for reverse body biasing. We therefore apply V_{DDH} to the gate of any PMOS

[1]There are many alternative power management interfaces and the full range of possibilities is beyond the scope of this paper. For example, a macro may have internal logic that detects its own standby state and sends a standby request to the PMU. The PMU may then acknowledge the request, depending on the configuration of the whole system. The same logic can be used to detect the wakeup condition and to interface with the PMU to achieve a return to active mode.

Proceedings of the 8th International Symposium on Quality Electronic Design (ISQED'07)
0-7695-2795-7/07 $20.00 © 2007 IEEE

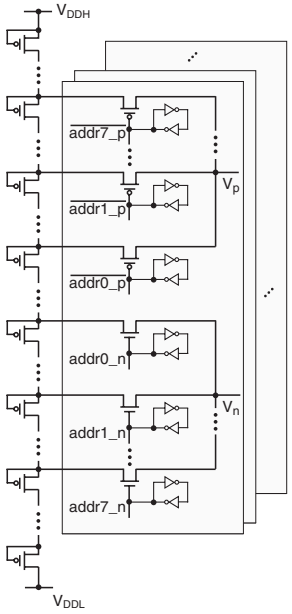

Fig. 3. Resistor tree for generating active mode body bias voltages.

switches that need to be turned off. Similarly, an NMOS switch is used to produce the NMOS body bias voltage, and we apply V_{DDL} to the gates of switches that are to be turned off. For instance, suppose that `macro 1` in Fig. 1 makes the transition from standby to active mode. The PMU fetches the codeword `001`, which is then decoded to yield `01000000`. The logic level is shifted (see Fig. 2) so that, if the address is to be used for PMOS switches (see Fig. 3), `addr1` corresponds to V_{DDH} while the remaining bits correspond to V_{SS}; but if the address is destined for NMOS switches, `addr1` corresponds to V_{DD} while the remaining bits correspond to V_{DDL}.

The body of each PMOS device in the resistor tree is biased to its own source, meaning that the n-well of each device needs to be isolated. This represents an area overhead, but frees the PMOS devices from the body effect. It also guarantees the stability of bias voltages generated by the resistor tree, even if V_t changes. In other words, the bias voltages are determined only by the number of serially connected PMOS devices, and are not affected by process variations. This is an important property of a body bias controller.

Since we use the same resistor tree to bias all n macros, each macro uses a dedicated switch, as shown in Fig. 3. When the resistor tree is used to bias one of the macros, the status of the switches for all the other macros must be maintained, and this is achieved by latches at the gate input of all switches.

2) Amplifier: The PMOS devices in the resistor tree operate in the subthreshold region. Therefore, the current that they draw is the subthreshold leakage current, which is very small and inadequate to drive the body of a macro. An amplifier, as shown in Fig. 4, is therefore required to boost the weak current from the resistor tree for NMOS body biasing[2].

[2]A similar amplifier is used for PMOS body biasing. The polarity of all transistors is inverted and the control signals (`amp_on`, `standby`, and `wakeup`) are complemented. The supplies are V_{SS} and V_{DDH} instead of V_{DD} and V_{DDL}.

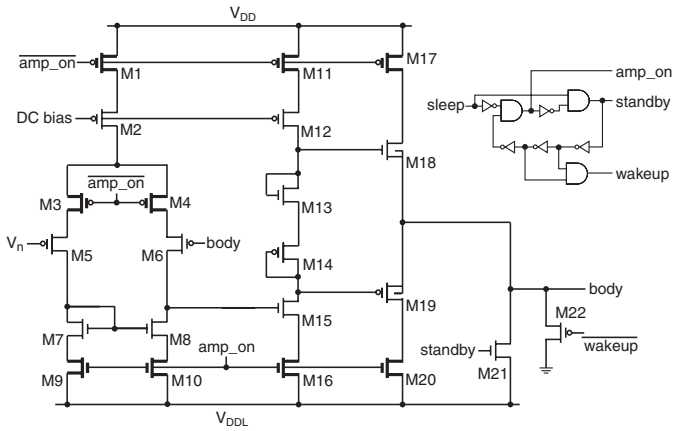

Fig. 4. Body bias amplifier.

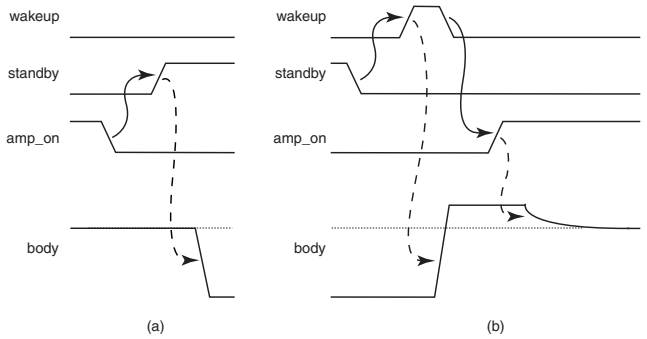

Fig. 5. Mode transition: (a) active to standby and (b) standby to active.

A simple two-stage amplifier is used: the first gain stage is a differential-input single-ended output stage, and the second is a common-source stage. Fig. 5(a) shows the control signals applied to the amplifier for the transition from active to standby mode. The `amp_on` signal is de-asserted first, which turns off the transistors highlighted in Fig. 4, so as to reduce the overall power consumption of the amplifier during standby mode. This is followed by asserting the `standby` signal, which turns on M21. This transistor then applies the predetermined large reverse body bias (V_{DDL}) to the bodies of the NMOS devices in a macro. Note that M22 remains turned off by the de-asserted `wakeup` signal. The presence of M3 and M4 is important for the safe operation of the amplifier. Since the gate of M6 is connected to the bodies of the NMOS devices in a macro, a large reverse body bias applied through M21 might reduce V_n, the output of the amplifier, at the gate input to M5. This would affect the potential of the resistor tree in the opposite direction, which might in turn affect the body bias of other macros in active mode, since the one resistor tree is shared among all macros. This potential problem can be avoided by turning off M3 and M4, which cuts the path from M6 to M5.

Fig. 5(b) shows the transition from standby to active mode. The `standby` signal is de-asserted, which turns off M21. M22 is then turned on by `wakeup`, and the body potential of NMOS devices quickly goes up from V_{DDL} to V_{SS}. Once the

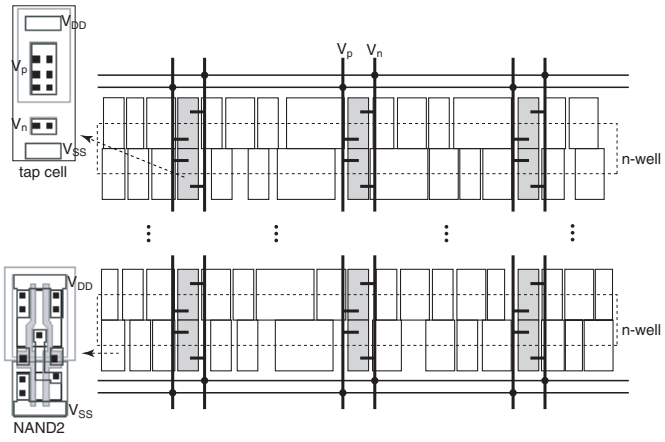

Fig. 6. Layout methodology.

body is stable at V_{SS}, M22 is turned off, and the amplifier is subsequently turned on by the `amp_on` signal. The bodies of the NMOS devices gradually settle down to the potential that is required to compensate for the process variation of their macro. The presence of M22 is also important in the transition from standby to active mode. If we switch directly from a large reverse body bias to an active-mode body bias, which is around V_{SS} for NMOS devices, the potential at the gate of M6 can affect the gate potential of M5. We alleviate this problem by using M22 to boost the body potential from V_{DDL} to V_{SS}, and then turn on the amplifier by means of the `amp_on` signal.

The circuit that generates the control signals (`wakeup`, `standby` and `amp_on`) from the `sleep` signal received from the PMU is also shown in Fig. 4.

C. Design Methodology for Cell-Based Semicustom Design

In order to validate the proposed lookup table-based adaptive body biasing in semicustom designs using standard-cell elements, we developed a custom cell library and associated layout methodology. We took 21 cells (four inverters, three 2-input NAND gates, one 3-input NAND gate, one 4-input NAND gate, one 2-input NOR gate, one tri-state buffer, six flip-flops, and four latches) from a commercial 180nm cell library, removed the body contacts, optimized the layout, and then re-characterized the devices using SPICE simulations. By optimizing the layout, we were able to reduce the height of each cell by 11%, which achieves a saving of area. Note that these tap-less cells are becoming popular in nanometer technologies such as 65nm.

Our layout methodology is shown in Fig. 6. A new tap cell [7] was designed to deliver the body biases, supplied by the adaptive body bias controller, to the n-well and p-well. The tap cells are inserted in a regular fashion as shown in Fig. 6. They are fixed in their locations, and then the logic cells are placed and routed automatically. The columns of the tap cells are separated by 50μm [7]. The layout of a tap cell and of a 2-input NAND gate are also shown in the figure. The application of this layout methodology to example circuits will be demonstrated in Section IV.

III. Optimization of Bias Voltages

The resistor tree generates a fixed number of predetermined body bias voltages. So, if a certain macro needs a bias voltage which is not generated by the resistor tree, the voltage that is closest (and larger in the case of NMOS, or smaller in the case of PMOS) to the required voltage has to be used. For example, suppose a macro is too slow due to process variation, but compensation can be achieved by applying 12mV to the body of its NMOS devices. Suppose also that the resistor tree generates 10mV and 20mV among its bias voltages. We need to apply 20mV since 10mV would not meet the delay target of the macro because it would still be too slow. But at 20mV, the macro is too fast and too leaky. Thus we see that it is important to determine a set of body bias voltages which minimize the overall excessive active leakage.

We can model the change in threshold voltage due to process variation using a random variable x that follows a normal distribution [2]:

$$x \sim N(\mu, \sigma^2), \tag{1}$$

where μ is the mean (i.e. the threshold voltage corresponding to a perfect process) and σ denotes the standard deviation. Let $x_1, x_2, \ldots, x_n$ be n threshold voltages and let $V_1, V_2, \ldots, V_n$ be n body bias voltages, such that, by applying V_i to the body of an NMOS transistor with a (zero-bias) threshold voltage of x_i, its threshold voltage becomes equal to μ. This implies that V_i is negative (reverse body bias) if $x_i < \mu$ (device is too fast), and positive (forward body bias) otherwise. A similar analysis applies to PMOS devices, but we will focus on NMOS in this section. We want to choose a set of body bias voltages, one of which can compensate for any threshold voltage variation, while the excessive active-mode leakage is minimized overall.

The subthreshold leakage of a turned-off NMOS device in a CMOS inverter can be approximated by

$$I_{sub} \propto e^{-V_{th}/S}, \tag{2}$$

where S denotes the subthreshold slope. As a measure of over-compensation (i.e. the amount of excessive active mode leakage), we let

$$Q = \int_{-\infty}^{+\infty} \left[e^{-V_t(x)/S} - e^{-\mu/S} \right] f_x(x)\, dx, \tag{3}$$

where $V_t(x)$ is the compensated threshold voltage of an NMOS device whose original threshold voltage was x, and where $f_x(x)$ represents the Gaussian probability density function, as shown in Fig. 7.

Now suppose we have an NMOS device with a threshold voltage between x_2 and x_3. We need to apply V_3, since V_2 would still makes the device too slow. Since the device is now too fast, its leakage is excessive. The extent of this leakage is determined by the new threshold voltage resulting from the application of V_3, which cannot be expressed by a closed-form

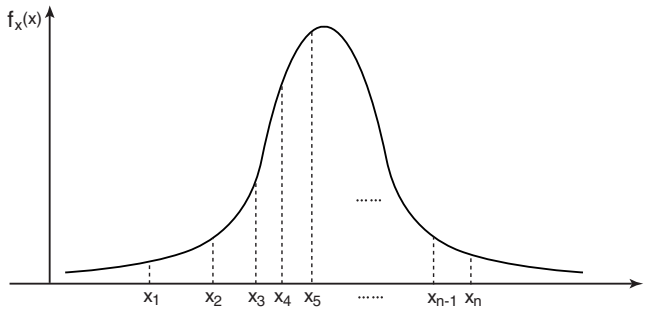

Fig. 7. Normal density function describing threshold variation.

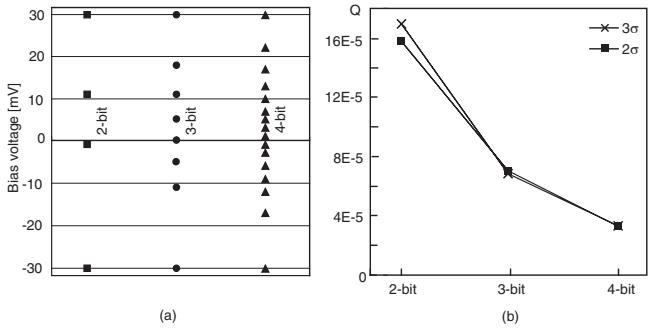

Fig. 8. (a) Optimized body bias voltages and (b) variation of excessive leakage with number of bits in the codeword.

equation. Instead, we approximate this new threshold voltage by $\mu - (x_3 - x)$, so that

$$V_t(x) = \mu - (x_i - x), \quad x_{i-1} < x < x_i. \tag{4}$$

Substituting (4) into (3) yields

$$\begin{aligned} Q &= e^{-\mu/S} \int_{-\infty}^{x_n} \left[e^{(x_i - x)/S} - 1 \right] f_x(x)\, dx \\ &\approx e^{-\mu/S} \left[\int_{-\infty}^{x_n} e^{(x_i - x)/S} f_x(x)\, dx - 1 \right]. \end{aligned} \tag{5}$$

Note that the upper limit of the integration is x_n and not ∞, since it is not possible to compensate for threshold voltages larger than x_n. This is not a problem in practice because we can use a large value, such as 3σ, for x_n, which enables us to compensate for all likely threshold variations.

Once we have μ and σ for the process variation and S, which is the subthreshold slope, we can minimize Equation (5), which gives us values for $x_1, x_2, \ldots, x_n$. This in turn yields a set of body bias voltages $(V_1, V_2, \ldots, V_n)$, which we can use to design our resistor tree. As an example, for a commercial 180nm CMOS process, $\mu = 400$mV and the 3σ variation is 30mV for NMOS devices. We vary the number of bits in the codewords in the lookup table (see Fig. 1) from 2 to 4, giving 4, 8, or 16 body bias voltages. In all cases, we fix the values of x_1 and x_n to -3σ and 3σ, which corresponds to minimizing Equation (5) for $2^n - 2$ bias voltages with an n-bit codeword.

We used Maple to minimize Equation (5), and Fig. 8(b) shows the results at room temperature (Q varies with S, which is a function of temperature). The figure clearly shows that, as we increase the number of bits in the codeword, the overall excessive leakage goes down, as we expect, since the increased number of available bias voltages allows finer control of body bias. However, the difference in leakage between 3-bit and 4-bit codewords is not significant, and it would therefore be reasonable to choose a 3-bit codeword because using 4 bits increases the complexity of the controller significantly (see Figs. 1 and 2). Fig. 8(a) shows the bias voltages that were obtained as a result of this optimization process. We repeated this experiment, with $x_1 = -2\sigma$ and $x_n = 2\sigma$. The results of this test are also shown in Fig. 8(b), but the difference between 2σ and 3σ is not significant. Additionally, we repeated the same experiment for different temperatures, but the results were virtually unchanged, implying that Q in our definition is not a strong function of temperature.

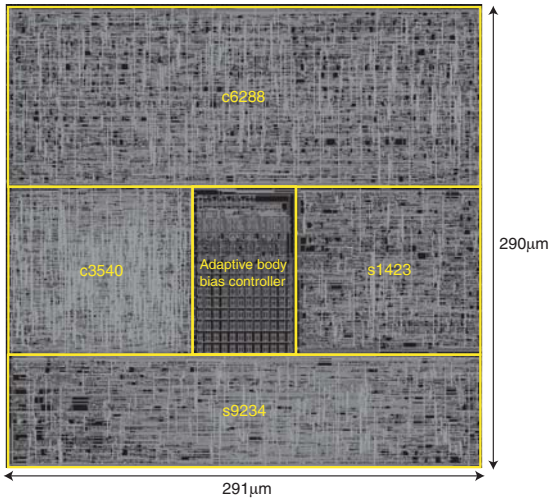

Fig. 9. The layout of four benchmark circuits and adaptive body bias controller.

IV. Experimental Results

We performed experiments on a set of four circuits taken from the ISCAS'89 benchmarks. Table I gives the characteristics of the original circuits. Each circuit was mapped on to a commercial 180nm triple-well, 1.8V gate library. Using the same 21 gates from the library, we were able to compare the original circuit with the one that is mapped to our custom library. Each circuit was placed and routed, and used the area shown in the third column. The transistor-level netlist is then extracted from the layout and simulated to determine the standby leakage current and the active-mode circuit delay.

The sixth column of Table I shows the area of each circuit when mapped on to our custom cell library, as explained in Section II. Compared to the original circuit, the use of custom cells gives us area savings of between 7% and 11% even including tap cells, due to the reduced cell height. The layout of a design consisting of four circuits together with an adaptive body bias controller is shown in Fig. 9. The controller occupies an area of $70\mu m \times 105\mu m$, of which 57% is taken up by the resistor tree. The size of this proportion is due to the well

TABLE I

EXPERIMENTAL RESULT ON ISCAS BENCHMARK CIRCUITS AT ROOM TEMPERATURE, FOR V_{DD} = 1.8V

Circuits	Original circuit				Test circuit				
	Gates	Area (μm^2)	Leakage (nA)	Delay (ns)	Area (μm^2)	ΔV_t (mV)	Leakage (nA)	Compensated delay (ns)	V_n (V) / V_p (V)
c3540	1669	120×105	512	2.304	107×109	-30	4.12	2.291	+0.2 / +1.6
c6288	2416	315×115	910	0.813	291×112	-10	13.43	0.812	+0.1 / +1.7
s1423	731	121×105	170	3.135	110×107	10	3.76	3.134	-0.05 / +1.85
s9234	5808	315×70	1001	0.763	291×71	30	24.59	0.759	-0.15 / +1.95

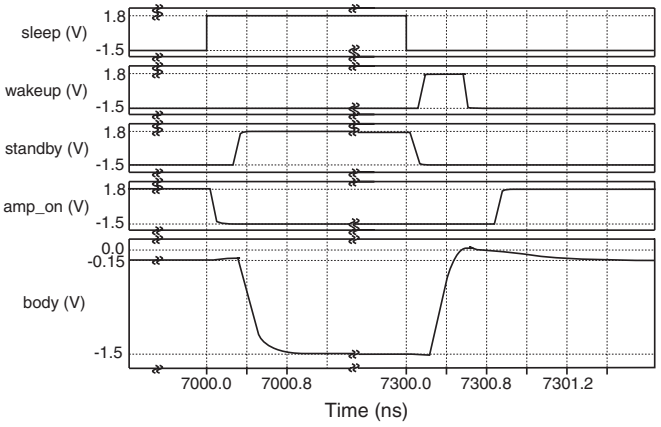

Fig. 10. Simulation of mode change from s9234: (a) active to standby and (b) standby to active.

isolation required by PMOS devices. The resistor tree consists of 96 PMOS devices, V_{DDH}=3.3V, and V_{DDL}=-1.5V, so that bias voltages between V_{DDL} and V_{DDH} can be generated in steps of 50mV. The negative voltage of V_{DDL} could be provided from out of the chip or could be generated by using a charge pump, which is beyond the scope of this paper. The codeword consists of 3 bits which gives good process compensation, as explained in the previous section.

In order to simulate the effects of process variation, we assumed that each circuit has a threshold voltage which differs from its standard value as shown in the seventh column of Table I. In the eighth column is the standby leakage current of each circuit. Compared to the original circuit, the leakage is cut by a factor of between 40 and 124, due to the large reverse body bias that we use in standby mode. The ninth column shows the delay in each circuit when active mode body bias is applied; the amount of bias is shown in the last column of the table. In contrast with the delays in the original circuit, all the circuits are now compensated.

Fig. 10 is a SPICE simulation result of a mode change from s9234. The `sleep` signal received from the PMU is shown at the top of the figure, while the three subsequent plots show the control signals that are generated from it. The body potential of NMOS devices in s9234 is also shown in the figure. It is readily seen that both transitions take about 1ns, which is short enough for fast mode change.

V. CONCLUSION

An adaptive body biasing has been used to compensate for process variation and to reduce subthreshold leakage current. The overhead of biasing circuits has limited its use to chip-level. In this paper, we have proposed a new adaptive body biasing scheme that can be used in block by block basis. The proposed scheme uses a lookup table that holds a codeword corresponding to the active mode body bias of each block on a chip, which, when applied, can compensate for process variation. A predetermined reverse body bias is used to reduce subthreshold leakage in standby mode. Since a fixed number of predetermined bias voltages are used, it is important to design them in efficient way, which we formulated and solved in a numerical fashion. We have presented the layout methodology for applying the proposed scheme to semi-custom designs using standard-cell elements. We performed an experiment with benchmark circuits, and have demonstrated that, through the use of proposed scheme, process variations can be compensated for and standby leakage current is reduced significantly.

ACKNOWLEDGMENT

This work was supported by Samsung Electronics.

REFERENCES

[1] T. Kobayashi and T. Sakurai, "Self-adjusting threshold-voltage scheme (SATS) for low-voltage high-speed operation," in *Proc. Custom Integrated Circuits Conf.*, May 1994, pp. 271–274.

[2] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi, and V. De, "Parameter variations and impact on circuit and microarchitecture," in *Proc. Design Automat. Conf.*, June 2003, pp. 338–342.

[3] S. G. Narendra and A. Chandrakasan, Eds., *Leakage in Nanometer CMOS Technologies*, Springer, 2005.

[4] J. W. Tschanz, S. G. Narendra, Y. Ye, B. A. Bloechel, S. Borkar, and V. De, "Dynamic sleep transistor and body bias for active leakage power control of microprocessors," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 11, pp. 1838–1845, Nov. 2003.

[5] T. Kuroda, T. Fujita, S. Mita, T. Nagamatsu, S. Yoshioka, K. Suzuki, F. Sano, M. Norishima, M. Murota, M. Kako, M. Kinugawa, M. Kakumu, and T. Sakurai, "A 0.9-V, 150-MHz, 10-mW, 4 mm^2, 2-D discrete cosine transform core processor with variable threshold-voltage (vt) scheme," *IEEE Journal of Solid-State Circuits*, vol. 31, no. 11, pp. 1770–1779, Nov. 1996.

[6] J. W. Tschanz, J. T. Kao, S. G. Narendra, R. Nair, D. A. Antoniadis, A. P. Chandrakasan, and V. De, "Adaptive body bias for reducing impacts of die-to-die and within-die parameter variations on microprocessor frequency and leakage," *IEEE Journal of Solid-State Circuits*, vol. 37, no. 11, pp. 1396–1402, Nov. 2002.

[7] L. T. Clark, M. Morrow, and W. Brown, "Reverse-body bias and supply collapse for low effective standby power," *IEEE Trans. on VLSI Systems*, vol. 12, no. 9, pp. 947–956, Sept. 2004.