Dual-Threshold Voltage Techniques for Low-Power Digital Circuits

James T. Kao and Anantha P. Chandrakasan, *Member, IEEE*

***Abstract*—Scaling and power reduction trends in future technologies will cause subthreshold leakage currents to become an increasingly large component of total power dissipation. This paper presents several dual-threshold voltage techniques for reducing standby power dissipation while still maintaining high performance in static and dynamic combinational logic blocks. MTCMOS sleep transistor sizing issues are addressed, and a hierarchical sizing methodology based on mutual exclusive discharge patterns is presented. A dual-V_t domino logic style that provides the performance equivalent of a purely low-V_t design with the standby leakage characteristic of a purely high-V_t implementation is also proposed.**

***Index Terms*—Domino logic, dual threshold voltage, low-power, MTCMOS, subthreshold leakage current, V_t.**

I. Sources of Power Dissipation in CMOS Circuits

In MODERN digital integrated circuits, power consumption can be attributed to three main components: short circuit, leakage, and dynamic switching power. Dynamic switching power is the dominant component of power consumption in modern integrated circuits, and results from the charging and discharging of gate capacitances during signal switching given by

$$P_{\text{switching}} = C_{\text{switched}} V_{\text{CC}}^2 f_{\text{clk}} \tag{1}$$

where C_{switched} is the total effective switched capacitance, V_{CC} is the supply voltage, and f_{clk} is the switching frequency. However, as scaling trends continue in future generations and as low-power voltage scaling becomes more aggressive, subthreshold leakage currents will become a larger, and potentially a dominant, component of overall power dissipation. Subthreshold leakage currents vary exponentially with threshold voltage and is given by

$$I_{\text{leakage}} = \frac{W}{W_0} I_0 e^{(Vgs - V_t)/nV_{\text{th}}} = \frac{W}{W_0} I_0 10^{(V_{gs} - V_t)/S} \tag{2}$$

where V_{th} is the thermal voltage, W is width, n is a constant, and $S = nV_{\text{th}}$ln 10 is the subthreshold slope. Thus, for a typical technology with a subthreshold slope of 100 mV/decade, each 100-mV decrease in V_t will cause an order of magnitude increase in leakage currents.

A. Scaling Impact on Leakage Currents

Technology scaling is one of the driving forces behind the tremendous improvement in performance, functionality, and power in integrated circuits over the past several years. With standard constant field scaling of 30%, one can expect frequency and switching power dissipation to scale by 50% each generation. However, for a constant die size, overall power dissipation due to dynamic switching currents remains relatively constant with scaling because the number of switching elements for the same die size will also increase by a factor of 50%. On the other hand, leakage currents increase exponentially. Although subthreshold leakage currents are not the dominant component of power dissipation in modern CMOS circuits, one can see that as a function of scaling the increase in leakage power can outpace dynamic switching power in future technologies [1].

Another major thrust in integrated circuit design is to minimize power dissipation while still maintaining high performance operation. From an energy efficiency point of view there is much potential to scale supply voltages to reduce power, but in order to maintain performance one must scale threshold voltages as well to maintain a large enough gate overdrive as shown in (3):

$$T_{pd} \propto \frac{CV_{\text{CC}}}{(V_{\text{CC}} - V_t)^{\alpha}} \tag{3}$$

where α models short channel effects [2]. Initially by scaling both V_{CC} and V_t, the increase in subthreshold leakage power will be small compared to the quadratic reduction in dynamic power supply in modern CMOS technologies. With extreme V_{CC} and V_t scaling however, the increase in leakage current will start to dominate the reduction in switching energies, indicating there must be an optimum V_{CC} and V_t point for a given target frequency. However, the optimal minimum energy V_{CC} and V_t point is significantly below the typical threshold voltage levels of today's technologies [3], [4].

II. Standby Leakage Current Reduction

From a technology scaling point of view, subthreshold leakage currents will continue to become a larger component in overall power dissipation. Likewise, from an optimal power point of view, the optimum energy point for V_{CC} and V_t during active modes will also correspond to a larger subthreshold leakage component. Leakage currents are especially important in burst mode type integrated circuits where the majority of the time the system is in an idle, or sleep, mode where no computation is taking place. For example, a cell phone, pager, or

Manuscript received January 14, 2000; revised March 1, 2000. This work was supported by Defense Advanced Research Projects Agency under Contract DABT63-95-C-0088 and under Contract SRC Task 633.001.

The authors are with the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139 USA.

Publisher Item Identifier S 0018-9200(00)04032-4.

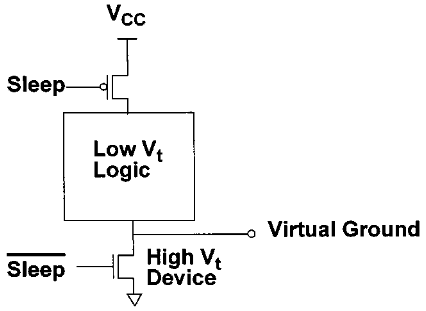


Fig. 1. MTCMOS circuit structure.

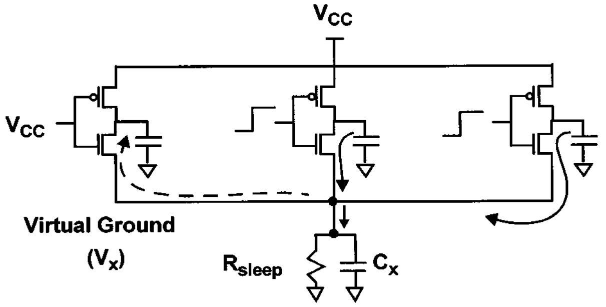


Fig. 2. MTCMOS block illustrating equivalent resistance, capacitance, and reverse conduction effects.

even an X-terminal will spend upwards of 90% of its time in a standby mode where the processor is waiting for user input. For this class of burst-mode-type applications, it may be acceptable to have large leakage currents during the active mode, but it is extremely wasteful to have large leakage currents during the idle state because power will be continuously drained with no useful work being done.

There have been several proposed techniques to help reduce subthreshold leakage currents during standby modes. Examples include utilizing the stack effect, where one can reduce subthreshold leakages by forcing series transistors to be simultaneously off, or using reverse body biasing to increase threshold voltages [5]–[8]. However, the stack effect only gives limited reduction over leakage currents, and body biasing effectiveness reduces with technology scaling. Another approach that can be quite effective at controlling subthreshold leakage currents is to use dual-threshold voltage technology. High-V_t devices can be used to reduce leakage currents while low-V_t devices can be used whenever high performance is required. The most straightforward application of dual-V_t technology is simply to partition a circuit into critical and noncritical regions, and to only use fast low-V_t devices when necessary to meet performance goals [9]. This approach will reduce subthreshold leakage currents both in the active mode and the standby mode, but may provide limited leakage reduction if the circuit contains many critical paths. The rest of this paper will explore two other dual-threshold voltage circuits styles for reducing standby leakage currents in combinational logic blocks. MTCMOS is geared toward static logic design while dual-threshold domino logic is geared toward dynamic logic solutions.

III. Multithreshold Voltage CMOS Technology

MTCMOS (multithreshold CMOS) is a dual-V_t technology that is very effective at reducing leakage currents in the standby mode. This technique involves using high-V_t transistors to gate the power supplies of a low-V_t logic block as shown in Fig. 1. When the high-V_t transistors are turned on, the low-V_t logic is connected to virtual ground and power, and switching is performed through fast devices. However, by introducing an extra series device to the power supplies, MTCMOS circuits will incur a performance penalty compared to CMOS circuits, which worsens if the devices are not sized large enough. When the circuit enters the sleep mode, the high-V_t gating transistors are turned off, resulting in a very low subthreshold leakage current from $V_{\rm CC}$ to ground [10] [11]. Although both pMOS and nMOS gating transistors are shown in Fig. 1, only one polarity sleep device is actually required to reduce leakage if the logic block is purely combinational. NMOS sleep transistors are typically more effective because they have lower "on" resistances, and subsequently can be made smaller for the same current drive. MTCMOS circuits can achieve several orders of magnitude reduction in leakage currents through two effects. First, the total effective leakage width of the original CMOS circuit is reduced to the width of the single "off" nMOS transistor (provided it is smaller than the original pulldown width), and second, the increased threshold voltage results in an exponential reduction in leakage currents. If the sleep transistor is turned off even more strongly (reversed bias), even further leakage reduction can be achieved.

MTCMOS is a very attractive technique for reducing subthreshold leakage currents during standby modes because existing designs (especially combinational logic blocks) can easily be modified into MTCMOS blocks by simply adding high-V_t power supply switches. Furthermore, the processing required to provide an extra threshold voltage involves only an additional implant processing step. However, serious drawbacks to the widespread use of MTCMOS are that appropriate sleep transistor sizing becomes very difficult and that sequential circuits will lose data when the power transistors are turned off.

A. MTCMOS Transistor Sizing Impact on Performance

Sleep transistors connecting power lines to virtual power lines can be accurately modeled as linear resistors when sized appropriately. For a turned-on nMOS sleep transistor sized large enough to ensure reasonable performance, the virtual ground voltage will be close to actual ground, so V_{ds} for the high-V_t switch will be small, making the linear approximation very accurate. Correct high-V_t sleep transistor sizing is a key design parameter that affects the performance of MTCMOS circuits. If sized too large, then valuable silicon area would be wasted and switching energy overhead between sleep and active modes would be increased. On the other hand, if sized too small, then the circuit would be too slow because of increased resistance to ground [12]. During the active mode, only the high-to-low transition is degraded by an nMOS series switch, whereas the low-to-high transition is unaffected.

When an MTCMOS block like the one shown in Fig. 2 is discharging, and neglecting the parasitic capacitance C_x, any charge flowing out of the low-V_t block will flow through the series sleep device and induce a voltage drop V_x. This voltage drop has two effects: first, it reduces the gate drive from V_{CC} to $V_{CC} - V_x$, and second, it causes the threshold voltage of the pulldown nMOS to increase due to the body effect. Both changes result in a decrease in the discharging current, which slows the output high-to-low transition. To maximize performance, the series transistor should be made as large as possible, subject to area and switching overhead constraints. As one continues to scale V_{CC} to lower voltages, the effective resistance of the high-V_t sleep transistors will continue to increase due to reduced $V_{gs} - V_t$, and thus even larger size series devices will be required to provide a small enough resistance. One can also employ overdriving the gate to help turn on the sleep transistors more strongly.

The parasitic capacitance due to wiring and junction capacitances on the virtual ground line shown in Fig. 2 actually helps reduce the virtual ground line bounce by serving as a local charge sink or reservoir for current. However, having a large capacitance in itself does not offset the effects of a poorly sized sleep transistor. Since current is constantly switching through the sleep resistance of a complicated logic block, the parasitic capacitance would have to be prohibitively large to prevent an IR drop from developing over time. With a large time constant, it will also take longer for the virtual ground node to discharge back to ground if it does reach a large value. While capacitance on the virtual power does help reduce transient spikes in MTCMOS circuits, proper sleep transistor sizing is still of utmost importance [12].

As illustrated in Fig. 2, MTCMOS logic blocks can also suffer from a reverse-conduction phenomenon where current flows backward from the virtual ground through the low-V_t nMOS transistor and charges up the output capacitance, and vice versa for a pMOS sleep transistor. More specifically, in the nMOS case, the virtual ground node can rise above 0 V so that another gate, which is supposed to be low, can experience reverse conduction as the output voltage rises from 0 V to V_x. This charging current comes from the discharging current of other gates transitioning from high to low, where a fraction of the discharge current is actually bypassing the sleep transistor. As a result, the MTCMOS circuit is slightly faster because the V_x voltage drop is not quite as large as one would expect if all current flowed through the sleep transistor to ground. Another effect of the reverse conduction, which pins output low voltages to V_x, is that a gate charging from low to high would be faster since it is already precharged to V_x. The drawback is that the noise margins in the circuits are reduced, and in the worst case the circuit can fail logically.

B. Inverter Tree Illustrating MTCMOS Delay

Fig. 3 is a typical inverter tree structure implemented in an MTCMOS technology with an nMOS sleep transistor that can be used to illustrate the effects of sleep transistor sizing on ground bounce and performance. The $0 \rightarrow 1$ transition is especially slow because in the final stage, all nine inverters are discharging simultaneously, which causes the virtual ground line to bounce significantly. Fig. 4 shows the virtual ground transient and reveals a gradual rise when the first inverter is discharging and a sharper "bump" when the final stage is reached. The figure also shows how the output waveform slows down when the sleep transistor width is too small. Conversely, virtual ground transients for the input $1 \rightarrow 0$ transition are smaller, so performance degradation is less than the previous scenario.

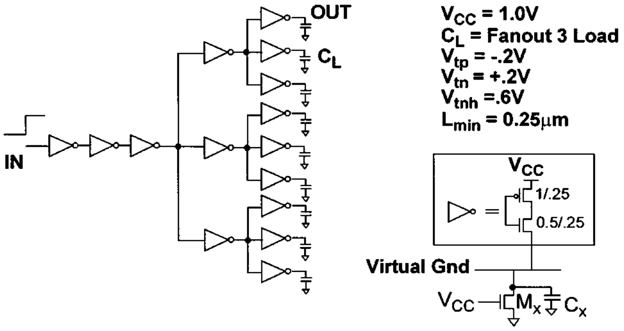


Fig. 3. MTCMOS inverter tree.

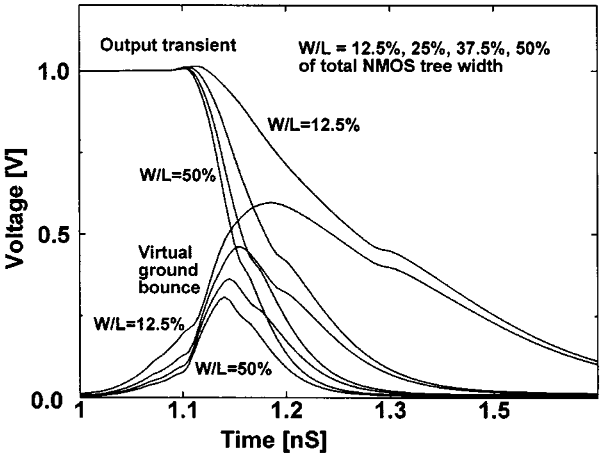


Fig. 4. Transient response for $0 \rightarrow 1$ transition.

C. Vector Dependency on MTCMOS Sizing

For more complex MTCMOS circuits, the input vector and resulting circuit discharge pattern plays a very important role in determining worst-case circuit performance. For example, the worst-case pattern for a base CMOS design will not typically translate to the worst- case pattern for an MTCMOS implementation because the MTCMOS circuit will be slowed down due to virtual ground bounce. Thus MTCMOS circuits will be more susceptible to input vectors that will cause large currents to flow through the sleep transistors, whereas ordinary CMOS circuits will not be affected. When analyzing MTCMOS circuits, one cannot simply examine the critical paths in the circuit, but must also consider all other accompanying gates that are switching. Because the worst-case delay is strongly affected by different input vectors and glitching behavior, it is very difficult to correctly size the sleep transistor. In fact, to optimally size the sleep transistor, one would need to exhaustively simulate the entire circuit for all possible input vectors and all sleep transistor sizes [12].

D. 8-Bit Carry-Save Multiplier Example

A larger MTCMOS circuit like an 8×8-bit carry-save multiplier demonstrates the impact of input vector on circuit performance. Because of size limitations, Fig. 5 shows only a 4×4

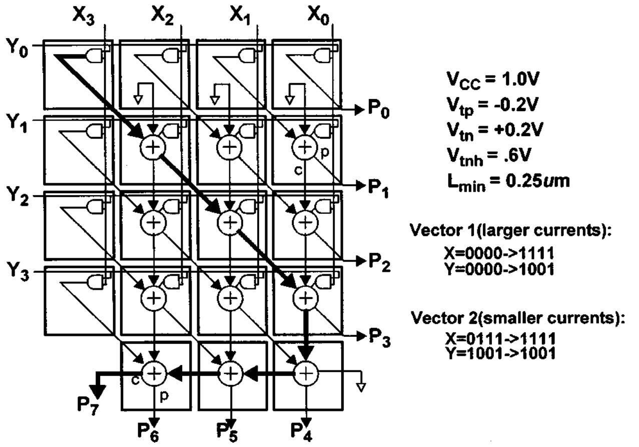


Fig. 5. Carry-save multiplier diagram (4 × 4 bit shown).

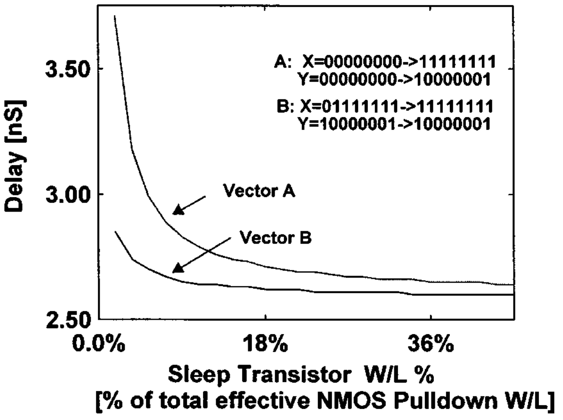


Fig. 6. 8×8 bit multiplier delay versus % W/L (shown as percentage of total effective nMOS pulldown W/L) for different input vectors (SPICE).

version with a worst-case delay path highlighted. Because of the regularity of this implementation, it is easy to see that one critical path (many others exist) lies along the diagonal and bottom row. However, two distinct input vectors that give the same delay in a CMOS implementation can give very different results in a MTCMOS circuit. The transition from $(x\colon 00, y : 00) \rightarrow (x : \mathrm{FF}, y : 81)$ for example causes many more internal transitions in adjacent cells and thus is more susceptible to ground bounce than the $(x\colon 7\mathrm{F}, y : 81) \rightarrow (x : \mathrm{FF}, y : 81)$ transition. The second input causes a rippling effect through the multiplier, where only a few blocks are discharging current at the same time. Fig. 6 shows how delay varies with the W/L ratio (expressed as a percentage of the total multiplier nMOS pulldown width) of the sleep transistor for these two cases.

Table I summarizes some key values from the plot. For example, if one wished to size the sleep transistor to provide less than 5% speed penalty for vector A, then one must size the sleep transistor W/L to be greater than 18% of the total effective W/L of the nMOS pulldown network for the multiplier. On the other hand, if one were to examine vector B, the same analysis could lead one to erroneously size the sleep transistor width to be only 5.4% of the total multiplier nMOS W/L, which would actually correspond to a 15.4% degradation in speed for the previous case. Since input vector strongly influences delays in MTCMOS, it is very important to determine the worst-case input vector for optimally sizing sleep transistors. However, to optimally size the sleep transistor would require exhaustive simulation of all possible input vectors, a task which is unrealistic for large systems.

TABLE I
CMOS DELAY, AND % DEGRADATION FOR VARIOUS W/L (SHOWN AS PERCENTAGE OF TOTAL CIRCUIT nMOS W/L) FOR TWO INPUT VECTORS

Initial X Y	Final X Y	Delay CMOS	% Degradation with W/L = 5.4%	% Degradation with W/L = 18%
0x 00 00	0x FF 81	2.59 ns	15.4%	4.6%
0x 7F 81	0x FF 81	2.58 ns	4.7%	1.6%

TABLE II
POWER/ENERGY CONSIDERATIONS FOR MTCMOS MULTIPLIER

Circuit Approach	Dynamic Switching Energy per Event	Leakage Power Active Mode	Leakage Power Sleep Mode	Sleep Switching Energy per Event	Sleep Switching Breakeven Time
CMOS	~ 4e-12 [J]	1.5e-4 [W]	1.5e-4 [W]	NA	NA
MTCMOS W/L 18%	~ 4e-12 [J]	1.45e-4 [W]	2.2e-9 [W]	3.1e-14 [J]	2e-10 [S]

The energy characteristics of the 8-bit multiplier are also summarized in Table II. For a standard low-V_t CMOS implementation, leakage power is a significant component of total power dissipation, but can be reduced almost five orders of magnitude by using high-V_t gating devices (sized 18% of the total W/L) during the standby mode. The switching energy required to go from sleep to active mode is small compared to the energy savings one could achieve during the low leakage standby state. For example, the sleep mode switching overhead energy would have been dissipated in only 200 pS during the high leakage condition. As a result, in this example it makes sense to place the multiplier in sleep mode even at fine-grain idle periods.

IV. HIERARCHICAL SIZING ALGORITHM BASED ON MUTUAL EXCLUSIVE DISCHARGE PATTERNS

Rather than searching for the worst-case input vector to exercise the worst-case discharge patterns in an MTCMOS combinational logic block, another approach is to synthesize an appropriate sleep transistor size based on mutual exclusive discharge patterns. Application of this sizing methodology will guarantee that the performance of a complex MTCMOS circuit will be within a chosen percentage of the original CMOS version for all possible inputs, but the sleep transistor would be larger than optimum [13].

This new sizing methodology ensures that the overall MTCMOS delay will be met by requiring that each individual gate will not degrade by more than a fixed percentage. For example, if one can guarantee that all elements degrade by no more than 5% during an MTCMOS implementation, then one can guarantee that any interconnection of MTCMOS gates will degrade by no more than 5% from its original CMOS counterpoint. Furthermore, if only a single polarity sleep transistor is used, then roughly only half of the individual MTCMOS gates will be degraded, resulting in an overall degradation of only 2.5% in performance for a balanced circuit. Forcing every single gate to meet a nominal performance measure is a much more demanding criteria than simply constraining the cumulative delay. However, in the context of MTCMOS circuits, it is much easier to implement this sizing strategy because one does not need to determine the worst-case input vector pattern for the whole circuit. Instead, each individual gate can be assigned it's own high-V_t sleep transistor, whose size can be locally determined through exhaustive simulations.

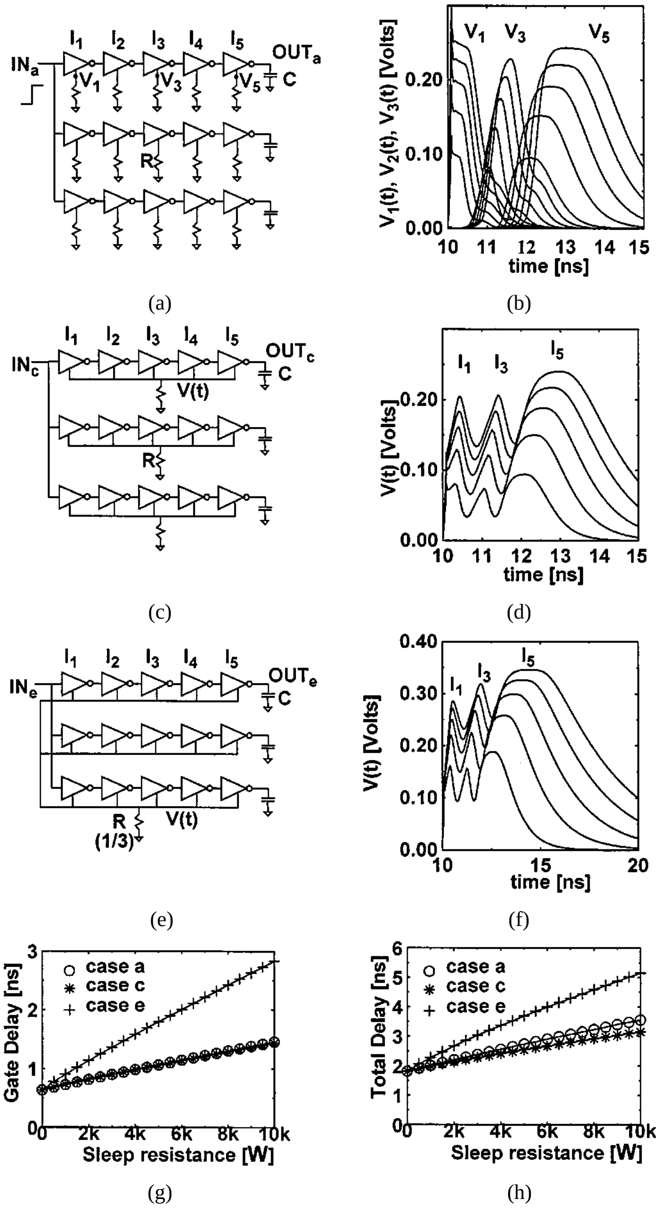
Once the MTCMOS circuit is sized with individual sleep transistors, one can then systematically merge the sleep transistors together because they can be shared among mutually exclusive gates, where no two gates can be discharging current at the same time. Finally, these sets of sleep transistors can then be combined to make a single sleep transistor for the whole circuit that guarantees that for any input vector, the MTCMOS circuit performance will be within the specified range of the corresponding CMOS circuit.

A. Example of Mutual Exclusive Sizing Technique

Fig. 7 shows a simple circuit consisting of three chains of five low-V_t transistors and illustrates how individually sized sleep transistors can be combined into a common power switch for a larger block of logic. Fig. 7(a) shows the first step in the transistor sizing procedure, where individual sleep resistors (which model sleep transistors in the "on" state) are sized to ensure that no gate degrades by more than a fixed percentage. The overall degradation of the series degenerated gates will be less than the individual gate degradation because the low-to-high transitions of I_2 and I_4 are not degraded by the nMOS sleep transistor. Fig. 7(b) shows how the virtual ground lines (V_1, V_3, and V_5) for this circuit will fluctuate as a result of a rising step function applied to the input.

Fig. 7(c) shows how the original inverter tree's sleep resistors can be replaced by only three resistors by utilizing the same high-V_t switch for mutual exclusive gates. Inverters I_1, I_2, I_3, I_4, and I_5 for example will never transition from high to low at the same times, and as a result can share a common sleep transistor. In general, for a set of n mutually exclusive gates with equivalent sleep resistances $r_1, r_2, \cdots, r_n$, the sleep resistors can be combined and replaced by a single $r_{\text{eff}} = \min(r_1, r_2, \cdots, r_n)$. As a result, the virtual ground bounce that each transitioning gate experiences will be the same, or smaller, than before. An added benefit of replacing n sleep resistors with a single one is that the subthreshold leakage current will decrease approximately by a factor of n, and also the increased parasitic capacitance on the virtual ground line can improve performance.

In Fig. 7(e), the three separate sleep resistors from Fig. 7(c) can be replaced by a single resistor with three times the conductance that now gates the entire circuit. Fig. 7(g) and (h) shows comparisons of the delay versus sleep resistor size for these two cases and illustrates how the resistance must be lowered by one-third in order to achieve the same performance. Another way to view this relationship is to examine the virtual ground transient response shown in Fig. 7(d) and (f). By scaling the resistance by one-third for the case with a single global sleep transistor, the virtual ground bounce shown in Fig. 7(f) can be matched to that of Fig. 7(g), which would give the same delay behavior. In general, combining separate sleep transistors into a single common one will be beneficial. The increased parasitic capacitances will tend to speed up the circuit during transient activity. Furthermore, because the larger resistances used in the original subcircuits are replaced by a smaller resistance applied to the combined circuit, in many cases individual gates will be faster than before.



(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 7. Inverter chain example showing the three steps for merging sleep resistors. Simulation parameters: $V_{CC} = 1.0$ V, $V_t = 0.2$ V, $C = 50$ fF, $1_{\min} = 0.7\,\mu$m. (a) Individual sleep resistors for each gate. (b) Virtual ground bounce for (a) $R = 2$, 4, 6, 8, and 10K. (c) Sleep resistor sharing for mutal exclusive gates. (d) Virtual ground bounce for (c) $R = 2$, 4, 6, 8, and 10K. (e) Sleep resistors combined through parallel combination. (f) Virtual ground bounce for (e) $R = 2$, 4, 6, 8, and 10K. (g) Delay of gate 15 alone. (h) Delay from input to output.

B. Sleep Transistor Sizing Algorithm

The previous example demonstrated how MTCMOS sleep transistors can be sized individually for each gate and then

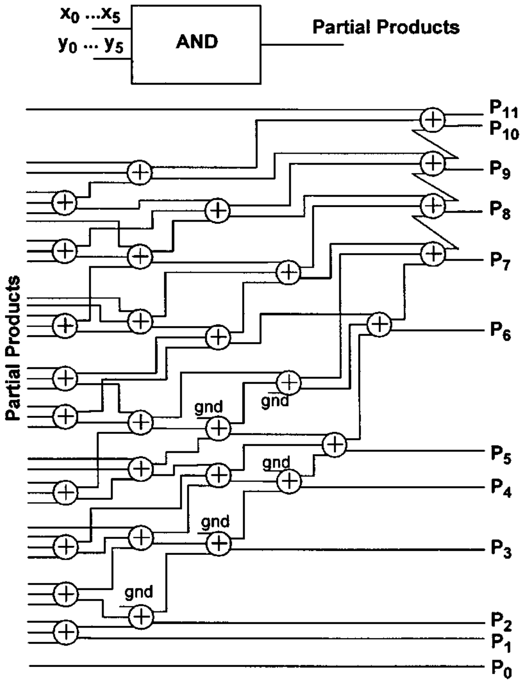


Fig. 8. 6 × 6 Wallace multiplier.

shared among mutually exclusive gates, where no two gates can be discharging current at the same time. The primary value of this technique is in the sleep transistor reduction step, because area of the sleep transistor is of primary concern in MTCMOS circuits. One approach to develop a mutual exclusive set of gates in a circuit is to use a criteria based on the structural interconnections of the network graph. Assuming a unit delay model for each gate, one can tabulate all possible times that any particular gate can switch. Mutually exclusive gates can then be grouped together whenever there is no intersection between corresponding sets of times. This merging technique based on mutual exclusive gate discharge patterns is most effective for balanced circuits with minimal glitching. Fortunately, a large class of circuits fall into this category, especially since less glitching is attractive from a low-power point of view [14].

This sleep transistor sizing methodology based on mutual exclusive discharge patterns is a generic technique applicable to LSI logic and was applied to a 6 × 6-bit Wallace tree multiplier shown in Fig. 8. The Wallace tree multiplier is a circuit that is well suited for this algorithm because there are many mutually exclusive gates that cannot transition at the same time. Initially, the AND gates and the carry-save adder units (with representative loadings) were simulated in SPICE to determine optimal high-V_t sleep transistor sizes (actually equivalent resistances) for each unit to give rise to a fixed degradation in performance. To achieve a degradation of 10% and 5%, the CSA required sleep transistors with equivalent resistances of 1600 Ω and 800 Ω, respectively. Likewise 10% and 5% degradation of the AND gates required equivalent resistances of 2700 Ω and 1350 Ω, respectively.

Next, the sleep transistor reduction and merging steps were performed to give rise to an equivalent resistor that could gate the entire multiplier. By tabulating all possible time periods that each cell can transition using the algorithm described above, we were able to reduce the 36 AND-cell and 30 adder-cell sleep resistors into 21 AND-cell and 15 adder-cell sleep resistors. The total equivalent resistance for the multiplier then corresponded to 60 Ω and 40 Ω for 10% and 5% maximum degradation. The merged resistance is a factor of two greater than the case where no merging takes place, which corresponds to a factor of two decrease in required sleep transistor width. The branches of this Wallace tree structure were not completely balanced because adder cells at inner levels of the tree could actually receive inputs from two levels prior. Another implementation that balances the paths more carefully would result in smaller sleep transistor sizes from the merging algorithm.

C. Hierarchical Sizing Methodology

Although the MTCMOS transistor sizing algorithm has been presented at the gate level, it can be applied at many hierarchical levels of a circuit. The algorithm simply operates on generic circuit blocks that are elements within a larger module, and each block is assumed to have a local high-V_t sleep transistor that is used for gating the power supply rails. These blocks can then be combined together using the mutual exclusive sizing algorithm described earlier. For example, blocks can represent individual gates, cells within an array (like adder cells in a multiplier), or even modules within a chip (like a shifter or adder that are mutually exclusive in an ALU). In all these cases, a gating sleep transistor can be shared among several different blocks if those blocks have activity patterns that do not overlap in time.

In order to achieve the best results, one should initially use a detailed simulator like SPICE to simulate as large a block as possible and to exhaustively determine the optimal sleep transistor size. Next, the hierarchical merging technique can then be applied to these existing blocks to synthesize an overall sleep transistor for the larger module, where determining a worst-case input vector would have been exceedingly difficult. Utilizing a hierarchical approach to sizing the sleep transistors is very attractive because detailed circuit complexity can be abstracted away at the expense of accuracy. One limitation of sharing a single sleep transistor among several distinct blocks is that one must also take into account the increased interconnect resistance for blocks that are far away from the sleep transistor. As a result, one may need to size sleep transistors larger than expected to compensate for the added interconnect resistances and may also need to widen the virtual ground wires to maintain performance.

V. DUAL-V_t DOMINO LOGIC

MTCMOS circuits require the insertion of extra series high-V_t devices which have no other purpose but to limit leakage currents during the standby mode. However, these sleep transistors are difficult to size correctly, and being in series with the pulldown–pullup path will always degrade performance. Another style of dual-threshold voltage design that addresses these issues is embedded dual-V_t logic, of which dual-V_t domino is a special case. In imbedded dual-V_t logic, high and low threshold voltages are assigned to the devices already existing in a logic gate, thereby eliminating the need for extra series switches. Furthermore, the transistor sizings of

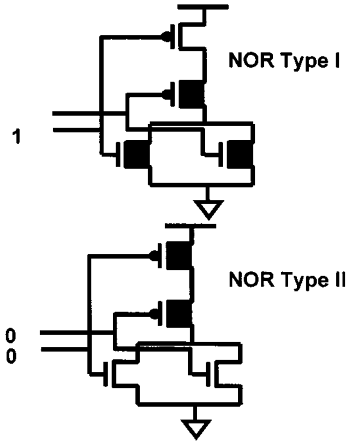


Fig. 9. Embedded dual-V_t NOR gates with low-V_t devices shaded. Inputs are shown to strongly turn off high-V_t devices for low standby leakage operation.

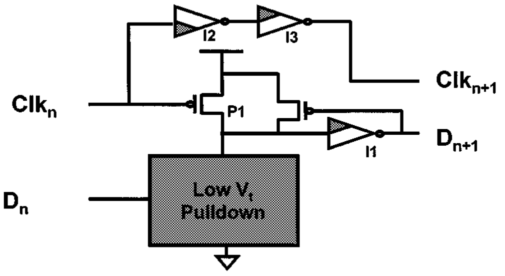


Fig. 10. Dual-V_t domino logic gate with low-V_t devices shaded.

the high-V_t devices are no longer impacted by the discharge patterns of other circuits, as was the case for MTCMOS. Fig. 9 shows two types of NOR gates in the imbedded dual-V_t circuit style, where existing devices are chosen to be high or low V_t, and gates can be placed in low leakage states. In order to compensate for reduced speed, the high-V_t devices need to be upsized, which can cause loading problems for previous stages. Furthermore, one must be able to configure each gate in the complicated circuit block with the correct input patterns during the standby mode, which may be difficult, if not impossible, for certain gate configurations. Fortunately, dual-threshold voltage domino solves both of these problems.

Dual-threshold voltage domino provides the performance equivalent of a purely low-V_t design with the standby leakage characteristic of a purely high-V_t implementation [15]. Because of the fixed transition directions in domino logic, one can easily place the dual-V_t domino gate into a low leakage state, and can imbed high-V_t devices in noncritical transition directions without impacting performance. In effect, the dual-V_t domino gate allows one to trade-off reduced precharge time for lower standby leakage currents. Dual-V_t domino methodology utilizes low threshold voltages for all transistors that can switch during the evaluate mode and utilizes high threshold voltages for all transistors that can switch during the precharge modes. Fig. 10 shows a typical dual-V_t domino stage, consisting of a pulldown network, inverter (I_1), leaker device (P_1), and clock drivers (I_2, I_3), with the low-V_t devices shaded.

Fig. 11 shows how this domino gate can be used in a typical pipe stage in a 2-phase clock-delayed domino methodology. The pipe stage shows a logic depth of 8 gate delays, consisting of four dynamic gates and four inverters, where the clock to downstream gates is tuned to match the flow of data. By delaying the clock with the data propagation, one can eliminate the need for nMOS footswitches in downstream domino gates (although it is still required for the first gate in the pipeline stage).

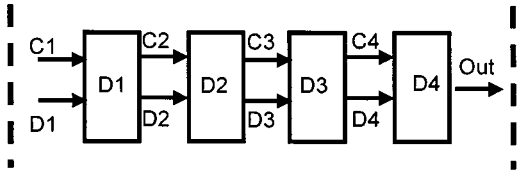


Fig. 11. Phi1 pipeline stage with four levels.

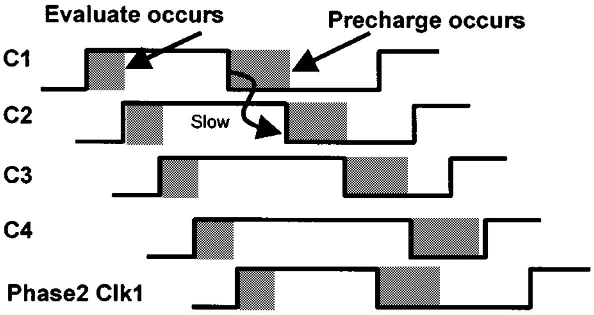


Fig. 12. Clocking methodology showing evaluate and precharge times.

A. Evaluate Mode

Before the domino gate in Fig. 10 enters the evaluate stage, the internal node is precharged high, while D_n, D_{n+1}, Clk_n, and Clk_{n+1} are all low. When Clk_n goes from low to high and data arrives on D_n, the domino gate will quickly evaluate through the low-V_t nMOS devices in the logic network and the low-V_t pMOS of I_1. Likewise the rising Clk_n signal will also pass through I_2 (fast pulldown) and I_3 (fast pullup) to supply the clocking signal to the next level of domino logic. The delay through I_2 and I_3 are matched to the delay through the logic and inverting stages such that the next data arrival is timed with the next evaluate clock. Finally, to maintain a high internal node voltage during evaluation, the P_1 transistor needs to supply enough current to satisfy the leakage from the low-V_t nMOS block. The main benefit of this dual-V_t domino approach however, is that during the evaluate phase, all transitions in the domino gate pass through low-V_t devices.

B. Precharge Mode

During precharge, the behavior of the circuit is the exact opposite, where the charging and discharging paths must pass through high-V_t devices. By balancing the clock drivers I_2, I_3 with the precharge time and I_1 delay, the data zeroing and clock precharge signal for the next stage will be closely aligned to avoid contention as illustrated in Fig. 12. Because high-V_t devices perform the precharge functions, the precharge time is longer than for the case where all low-V_t devices are used. As a result, the clock pulse width increases as one travels to downstream gates in order to maintain alignment between the precharge transition and clock. Since precharge time is not in the critical path, more time is available to finish the precharge

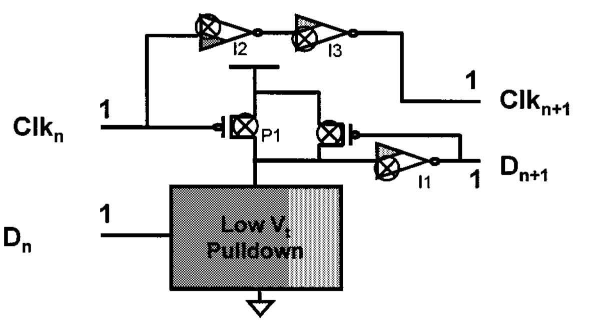


Fig. 13. Dual-V_t Domino gate in low leakage state.

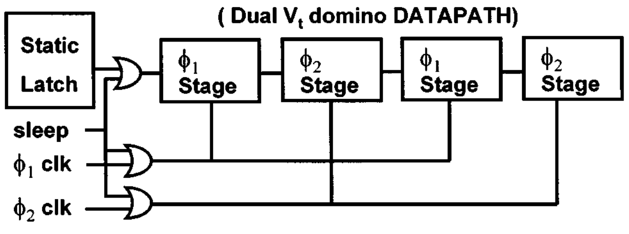


Fig. 14. Pipeline sleep mode circuitry.

transition, so using slower high-V_t devices is acceptable. Furthermore, in the case where the evaluate clocks are not 50% duty cycle, then even more time can be allotted for precharge. In the traditional domino style where all domino gates are clocked with the same clock (and nMOS evaluate switches are used to prevent contention), the entire precharge clock can be utilized for each gate.

C. Standby Mode

When a dual-V_t domino logic stage is placed in standby mode, the domino clock needs to be high (evaluate) in order to shut off the high-V_t devices. For example, the precharge pMOS device, the P_1 leakage device, the I_2 pMOS, the I_3 nMOS, and the I_1 nMOS all need to be turned off strongly in order to reduce leakage currents during the idle state, as shown in Fig. 13. Furthermore, to ensure that the internal gate node remains at a solid 0, the initial inputs into the domino gate must be set high. Otherwise, the internal node could float, and cause short circuit currents in the following inverter. All node voltages within the domino gate are thus actively driven during the standby mode, and all high-V_t devices are strongly turned off, yielding low subthreshold leakage currents.

Fig. 14 illustrates how to place a more complicated datapath consisting of several pipeline stages into standby mode. The first step is for the control circuitry to finish computing any instruction in the pipeline so that no data is lost. Next, both phases of the domino pipeline are placed in sleep mode by gating the clocks to a logic "1" so that all gates are in the evaluate mode. Last, the first level inputs to the beginning of the pipeline must also be gated to a logic "1," which will cause all subsequent gates in the pipeline to evaluate in a cascaded fashion. The resultant datapath will thus be in a low leakage state where all high-V_t devices are strongly turned off.

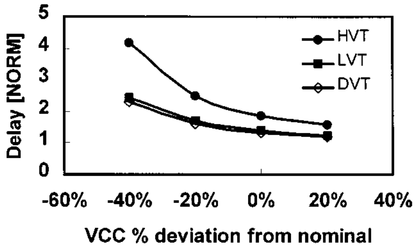


Fig. 15. Evaluation delay through pipeline stage. Delay (D rise to out rise).

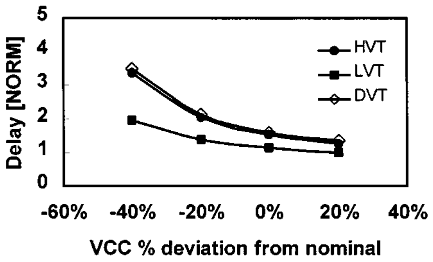


Fig. 16. Precharge delay for pipeline stage. Precharge (C1 Fall → Out Fall).

D. Simulation Results

To verify the functionality and benefit of dual-V_t domino logic, simulation were performed on a representative pipeline stage modeled as an inverter chain with four dynamic NOR gates and four accompanying static inverters in an aggressive 0.18 μm technology. The NOR gate has eight inputs, each driving a fanout of 3 load. These wide gates are a good representative of domino circuits, because domino technology is most effective for gates with wide, rather than deep, pulldown networks. The experimental circuit has the exact same structure as shown in Figs. 10 and 11. Simulations were performed on three circuit variants with the exact same transistor sizings: an all low-V_t design, an all high-V_t design, and a dual-V_t design. As predicted, the low-V_t delay is significantly faster than the high-V_t one. However, the dual-V_t has a fast evaluate time on par with the all low-V_t design, but has a slow precharge delay on par with the all high-V_t design.

The performance benefit of low-V_t domino is most apparent at lower voltages, where V_t is on par with V_{CC}. Fig. 15 shows a comparison of low-V_t, high-V_t, and dual-V_t delays as a function of the operating voltage, shown in the graph as a percent deviation from the nominal V_{CC} operating point. Clearly, the trend shows how low-V_t and dual-V_t benefits are most effective at low voltages. For example, at -40% deviation (low V_{CC}), the reduction in delay over a high-V_t implementation is 44.5%, while it is only 24.1% at $+20\%$ deviation (high V_{CC}). Interestingly, the dual-V_t circuit delay is actually faster than the all-low-V_t device in all cases, and this can be attributed to the fact that during switching, the pulldown network has less leakage contention from the off pMOS device in the dual-V_t case.

Fig. 16 on the other hand shows a plot of precharge delay as function of operating voltage. Precharge delay was measured as the delay between the falling Clk line at the input of the block to the falling edge (precharged state) of the final block output. As can be seen in the figure, the low-V_t implementation has a

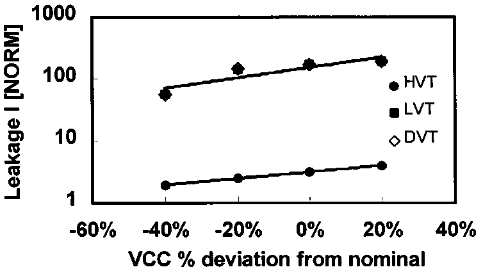


Fig. 17. Leakage current for $\mathrm{Clk} = 0$.

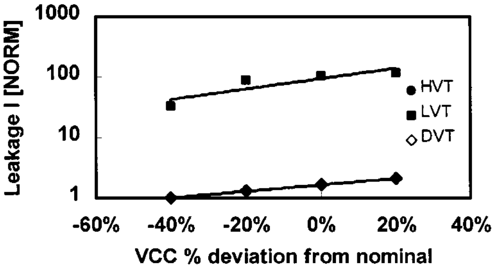


Fig. 18. Leakage current for $\mathrm{Clk} = 1$.

fast precharge delay, while the high-V_t and dual-V_t circuits have virtually identical but much larger delay times. Again, since the precharge delay is not in the critical path, this will not affect the overall circuit speed.

Simulations were also performed to verify the leakage benefits in the dual-V_t design. Two scenarios are explored: one where the circuit is stalled in the precharge mode with the data input zeroed, while the second scenario is where the circuit is stalled in the evaluate mode with all data inputs activated. As described earlier, the proper dual-V_t standby mode is the latter case.

Figs. 17 and 18 illustrate two different components of leakage reduction seen in the dual-V_t standby mode case. First of all, by holding the circuit in the evaluate mode rather than the precharge mode, the leakage will be reduced because the leakage path in each gate is through a single off pMOS, rather than eight off nMOS transistors in parallel. Thus leakage currents are reduced slightly in all three cases. For the dual-V_t case, the greatest benefit of holding the circuit in the evaluate mode is that the leakage path will be through a high-V_t pMOS device. As can be seen in Fig. 18, the dual-V_t implementation leakage is comparable to the leakage of the high-V_t implementation, both of which are an order of magnitude less than the low-V_t case. Another interesting phenomenon shown in the figures is the trend showing how low-V_t device leakage increases more rapidly than high-V_t devices with supply voltage. This scaling trend is due to worse short-channel effect on the low-V_t devices, making their leakage more susceptible to supply voltage scaling.

E. Dual-V_t Issues

In domino circuits, the noise margin is directly related to the threshold voltage of the nMOS pulldown tree, so there is definitely a limit to how low V_t's can scale. Furthermore, active leakage in large fan in gates, if large enough, can effect functionality when a domino gate tries to hold an internal node high. A large keeper device helps, but this will directly effect performance, and active leakage power dissipation still remains a problem. However, research has shown that domino gates can be made to function at low voltages and low V_t's. With careful attention to noise, the use of keeper devices, and improved device characteristics, domino logic will likely continue to be used in future technologies. As long as low V_t and low V_{CC} dynamic logic can be made to work, then it will be beneficial to use the dual-V_t domino methodology. Although it has little effect on active leakage power, dual-V_t domino significantly reduces standby leakage, which can play an important role in many applications where waiting times are long. Furthermore, switching to standby mode using this methodology has low overhead because one only needs to gate the clocks and then assert the initial inputs into the pipeline. As a result, this power down mode can also be effective at fine grain control such as for inactive modules within a chip like a multiplier or divider.

VI. Conclusion

This paper presented several dual-threshold voltage circuit techniques that can help reduce subthreshold leakage currents during standby modes for combinational logic blocks. MTCMOS was shown to be an effective standby leakage control technique for static logic, but difficult to implement since sleep transistor sizing is highly dependent on discharge patterns within the circuit block. A hierarchical transistor sizing methodology based on mutual exclusive discharge patterns was then presented that gives a computationally tractable, although not optimum, solution for MTCMOS sleep transistor sizing. This methodology provided an upper bound on the sleep transistor size needed to guarantee a specified performance level compared to the original CMOS counterpoint. Finally, a special case of imbedded dual-V_t applied to domino logic was presented, which took advantage of the fixed transition directions in domino logic to provide the performance benefits of an all low-V_t design yet still maintain the low subthreshold leakage characteristics of an all high-V_t design during the standby mode. Dual-V_t domino logic avoids the sizing difficulties and inherent performance penalty associated with MTCMOS, and can be used extensively throughout a domino datapath. Since subthreshold leakage currents will become an increasingly dominant component of overall power consumption in future technologies, dual-threshold voltage circuit techniques will play an important role in future circuit design.

Acknowledgment

The authors would like to thank S. Narendra and Prof. D. Antoniadis for their valuable input and feedback.

References

[1] V. De and S. Borkar, “Technology and design challenges for low power and high performance,” in *Proc. Int. Symp. Low Power Electronics and Design*, 1999, pp. 163–168.

[2] T. Sakurai and R. Newton, “Alpha-power law MOSFET model and it’s applications to CMOS inverter delay and other formulas,” *IEEE J. Solid-State Circuits*, vol. 25, pp. 584–594, Apr. 1990.

[3] A. Chandrakasan, I. Yang, C. Vieri, and D. Antoniadis, “Design considerations and tools for low-voltage digital system design,” in *ACM/IEEE Design Automation Conf.*, June 1996, pp. 113–118.

[4] R. Gonzalez, B. Gordon, and M. Horowitz, “Supply and threshold voltage scaling for low power CMOS,” *IEEE J. Solid-State Circuits*, vol. 32, pp. 1210–1216, Aug. 1997.

[5] M. Horiguchi, T. Sakata, and K. Itoh, “Switched-source-impedance CMOS circuit For low standby subthreshold current giga-scale LSI’s,” *IEEE J. Solid-State Circuits*, vol. 28, pp. 1131–1135, Nov. 1993.

[6] T. Kawahara, M. Horiguchi, Y. Kawajiri, G. Kitsukawa, and T. Kure, “Subthreshold current reduction for decoded-driver by self-reverse biasing,” *IEEE J. Solid-State Circuits*, vol. 28, pp. 1136–1144, Nov. 1993.

[7] Y. Ye, S. Borkar, and V. De, “A new technique for standby leakage reduction in high-performance circuits,” in *1998 Symp. VLSI Circuits*, June 1998, pp. 40–41.

[8] T. Kuroda and T. Fujita *et al.*, “A 0.9V, 150MHz, 10mW, 4mm^2, 2-DCT core processor with variable VT scheme,” *IEEE J. Solid-State Circuits*, vol. 31, pp. 1770–1778, Nov. 1996.

[9] W. Lee *et al.*, “A 1V DSP for wireless communications,” in *ISSCC*, Feb. 1997, pp. 92–93.

[10] S. Mutoh, T. Douseki, Y. Matsuya, T. Aoki, S. Shigematsu, and J. Yamada, “1-V power supply high-speed digital circuit technology with multithreshold-voltage CMOS,” *IEEE J. Solid-State Circuits*, vol. 30, no. 8, pp. 847–854, August 1995.

[11] S. Mutoh, S. Shigematsu, Y. Matsuya, H. Fukada, and J. Yamada, “1V multi-threshold CMOS DSP with an efficient power management technique for mobile phone application,” in *IEEE ISSCC*, 1995/1996, pp. 168–319.

[12] J. T. Kao, A. P. Chandrakasan, and D. Antoniadis, “Transistor sizing issues and tool For multi-threshold CMOS technology,” in *ACM/IEEE Design Automation Conf.*, June 1997, pp. 409–414.

[13] J. Kao, S. Narendra, and A. Chandrakasan, “MTCMOS hierarchical sizing based on mutual exclusive discharge patterns,” in *ACM/IEEE Design Automation Conf.*, June 1998, pp. 495–500.

[14] T. Sakuta, W. Lee, and P. Balsara, “Delay balanced multipliers for low power/low voltage DSP core,” in *IEEE Symp. Low Power Electronics*, 1995, pp. 36–37.

[15] J. Kao, “Dual threshold voltage domino logic,” in *25th Eur. Solid-State Circuits Conf.*, Sept. 1999, pp. 118–121.

James T. Kao received the B.S. degree in electrical engineering and computer science from the University of California, Berkeley, in 1993, and the S.M. degree in electrical engineering and computer science in 1995 from the Massachusetts Institute of Technology, Cambridge, where he is currently working toward the Ph.D. degree in the area of subthreshold leakage control techniques for low power digital circuits.

Anantha P. Chandrakasan (M’95) received the B.S., M.S., and Ph.D. degrees in electrical engineering and computer sciences from the University of California, Berkeley, in 1989, 1990, and 1994, respectively.

He has been at the Massachusetts Institute of Technology, Cambridge, since September 1994, and is currently an Associate Professor of electrical engineering and computer science. He held the Analog Devices Career Development Chair from 1994 to 1997. His research interests include the ultra low power implementation of custom and programmable digital signal processors, distributed wireless sensors, multimedia devices, emerging technologies, and CAD tools for VLSI. He is a co-author of the book titled *Low Power Digital CMOS Design* by Kluwer Academic Publishers, Norwell, MA, and a co-editor of the book *Low Power CMOS Design*, IEEE Press, New York, NY. He has served on the technical program committee of various conferences including ISSCC, VLSI Circuits Symposium, DAC, and ISLPED. He has served as a Technical Program Co-Chair for the 1997 International Symposium on Low-Power Electronics and Design (ISLPED), VLSI Design ’98,and the 1998 IEEE Workshop on Signal Processing Systems, and as a General Co-Chair of the 1998 ISLPED. He was the Signal Processing Sub-Committee chair for ISSCC ’00. He is also a member of the Design and Implementation of the Signal Processing Systems (DISPS) Technical Committee of the Signal Processing Society.

Dr. Chandrakasan received the NSF Career Development Award in 1995, the IBM Faculty Development Award in 1995, and the National Semiconductor Faculty Development Award in 1996 and 1997. He has received several best paper awards, including the 1993 IEEE Communications Society’s Best Tutorial Paper Award, the IEEE Electron Devices Society’s 1997 Paul Rappaport Award for the Best Paper in an EDS publication during 1997, and the 1999 Design Automation Conference Design Contest Award. He is currently an Associate Editor for the IEEE JOURNAL OF SOLID-STATE CIRCUITS.