A Predictably Low-Leakage ASIC Design Style

Nikhil Jayakumar and Sunil P. Khatri

***Abstract*—In this paper, we describe a new low-leakage standard cell based application-specific integrated circuit (ASIC) design methodology. This design is based on the use of modified standard cells, designed to reduce leakage currents (by almost two orders of magnitude) in standby mode and also allow precise estimation of leakage current. For each cell in a standard cell library, two low-leakage variants of the cell are designed. If the inputs of a cell during the standby mode of operation are such that the output has a high value, we minimize the leakage in the pull-down network, and similarly we minimize leakage in the pull-up network if the output has a low value. In this manner, two low-leakage variants of each standard cell are obtained. While technology mapping a circuit, we determine the particular variant to utilize in each instance, so as to minimize leakage of the final mapped design. We have performed experiments to compare placed-and-routed area, leakage and delays of this new methodology against Multithreshold CMOS (MTCMOS) and a regular standard cell based design style. The results show that our new methodology (which we call the "HL" methodology) has better speed and area characteristics than MTCMOS implementations. The leakage current for HL designs can be dramatically lower than the worst-case leakage of MTCMOS based designs, and two orders of magnitude lower than the leakage of traditional standard cells. An ASIC design implemented in MTCMOS would require the use of separate power and ground supplies for latches and combinational logic, while our methodology does away with such a requirement. Another advantage of our methodology is that the leakage is precisely estimable, in contrast with MTCMOS. Our primary contribution in this paper is a new low leakage design style for static CMOS designs. In addition, we also discuss techniques to reduce leakage in dynamic (domino logic) designs.**

***Index Terms*—CMOS integrated circuits (ICs), IC design, leakage currents, VLSI.**

I. Introduction

With diminishing process feature sizes and operating voltages, the control of (subthreshold) leakage currents in modern VLSI designs is becoming a significant challenge. Leakage currents increase exponentially with decreasing threshold voltages (which are typically maintained at a fixed fraction of supply voltage). This is expected to be a major concern for VLSI design in the nanometer realm [1].

The power consumed by a design in the standby mode of operation is due to leakage currents in its devices. With the prevalence of portable electronics, it is crucial to keep the leakage currents of a design small in order to ensure a long battery life in the standby mode of operation.

The leakage current for a pMOS or nMOS device corresponds to the $I_{\rm ds}$ of the device when the device is in the *cutoff* or *subthreshold* region of operation. The expression for this current [2] is

$$I_{\rm ds} = \frac{W}{L} I_0 e^{\left(\frac{V_{\rm gs}-V_T-V_{\rm off}}{nv_t}\right)} \left(1 - e^{\left(-\frac{V_{\rm ds}}{v_t}\right)}\right). \qquad (1)$$

Here I_0 and $V_{\rm off}$ (typically $V_{\rm off} = -0.08$ V) are constants, while v_t is the thermal voltage (26 mV at 300 °K) and n is the subthreshold swing parameter.

We note that $I_{\rm ds}$ increases exponentially with a decrease in V_T. This is why a reduction in supply voltage (which is accompanied by a reduction in threshold voltage) results in exponential increase in leakage.

Another observation that can be made from (1) is that $I_{\rm ds}$ is significantly larger when $V_{\rm ds} \gg nv_t$. For typical devices, this is satisfied when $V_{\rm ds} \simeq V_{\rm DD}$. The reason for this is not only that the last term of (1) is close to unity, but also that with a large value of $V_{\rm ds}$, V_T would be lowered due to drain induced barrier lowering (DIBL) (V_T decreases approximately linearly with increasing $V_{\rm ds}$) [3], [2]. Therefore, leakage reduction techniques should ensure that the supply voltage is not applied across a single device, as far as possible.

In recent times, much attention has been devoted to leakage current control [4]–[13]. These approaches employ devices with increased V_T values to reduce leakage. The modification of V_T is done either statically (at the time of device fabrication) or dynamically (by increasing V_T via body effect and bulk voltage modulation). We prefer the former for its simplicity of implementation.

The remainder of this paper is organized as follows. Section II discusses some previous work in this area. In Section III, we describe our method to control leakage currents in an arbitrary standard cell. In Section IV, we present experimental results comparing our idea with Multithreshold CMOS (MTCMOS) and with traditional standard cell-based ASIC design. In Section V, we discuss methods of reducing leakage in domino logic designs. Conclusions and future work are discussed in Section VI.

II. Previous Work

In recent times, leakage power reduction has received much attention in academic research as well as industrial applications. Several means of reducing leakage power have been proposed.

In [8], the authors propose a dynamic threshold MOSFET design for low leakage applications. In this scheme, the device gate is connected to the bulk, resulting in high-speed switching and low leakage currents through body effect control. The drawback of this approach is that it is only applicable in situations where $V_{\rm DD}$ is lower than the diode turn-on voltage. Also, the increased capacitance of the gate signal slows the device down, and as a result, the authors propose the use of this technique for partially depleted silicon on insulator (SOI) designs.

Manuscript received August 19, 2003; revised August 24, 2006.

The authors are with the Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77840 USA (e-mail: sunilkhatri@tamu.edu; nikhil@ece.tamu.edu).

Digital Object Identifier 10.1109/TVLSI.2007.893603

Another methodology for controlling leakage is the variable-threshold [6], [10], [11] (often called VTCMOS) approach. In such an approach, the device threshold voltages are controlled dynamically by modifying the device bulk voltage. This method offers the advantage of decreasing the leakage in standby mode while not increasing the delay in the active mode. The authors of [6] implemented a discrete cosine transform (DCT) core using this approach. However, complex control circuitry is required to generate and control the bulk voltages. Another drawback is that the bulk terminals have to be electrically isolated from the source terminals of the devices and this may require some major changes in cell layout. VTCMOS is not applicable to fully depleted SOI and is difficult to implement in partially depleted SOI [7]. In another approach called the super cut-off CMOS (SCCMOS) approach [7], the gate of a pMOS device which gates the $V_{\rm DD}$ supply is overdriven, thereby reducing the leakage dramatically. This again requires the design of complex circuitry to generate and control the special over-driven voltage values.

In [14], the authors suggest introducing a switched-impedance (a resistor for limiting the leakage current in parallel with a switch for bypassing the resistor) at the source of a MOS transistor. This switched-impedance is shared but this requires two additional power rails whose area overhead is not quantified. The results are given for only RAM circuits and not more general logic circuits. The authors of [15] introduce a scheme (for RAM circuits) with power supply gating transistors connected selectively to the sources of turned-off MOS transistors. The methodology introduced in [16] is similar to [15] but is applied in the context of SCCMOS. In [14]–[16], due to sharing of the power supply gating elements, the is an area overhead on account of two additional power rails. This additional area overhead is not quantified and compared with regular standard-cell based designs.

More traditional design approaches have suggested the use of dual-threshold devices [4] in an MTCMOS configuration MTCMOS utilizes nMOS and pMOS power supply gating devices. The authors propose a MTCMOS standby device sizing algorithm which is based on mutually exclusive discharging of gates. This technique is hard to utilize for random logic circuits as opposed to the extremely regular circuits which are used as illustrative examples in [4]. In [5], the authors describe an MTCMOS implementation of a phase locked loop (PLL) using a 0.5-μm process. In both these works, the problem of estimating the leakage of an MTCMOS design is not addressed. In practice, the leakage of such a design can vary widely and is hard to control or predict. The threshold voltage is modified by bulk bias (via body effect) and DIBL, which are determined in part by the voltages of the bulk/source and source/drain nodes. Since cell inputs and outputs as well as bulk nodes float in an MTCMOS design operating in standby mode, precise prediction or control of leakage is impossible in MTCMOS. The voltage of these floating nodes can significantly affect the device threshold voltages. Cell input and output voltages affect the leakage of a gate as seen in (1). The bulk voltage V_b affects V_T through body effect, and subthreshold leakage has an exponential dependence of V_T as seen in (1). The body effect equation can be written as $V_T = V_T^0 + \gamma\sqrt{V_{sb}}$, where V_T^0 is the threshold voltage at zero V_{sb}. This results in a situation where MTCMOS designs can have a large range of leakage currents, with little ability to predict the actual leakage current. Another drawback of MTCMOS is that memory elements in MTCMOS would require clean power supplies routed to them if we want to maintain their state in standby mode [5]. In [9], an MTCMOS-like leakage reduction approach was proposed, in which the MTCMOS sleep devices are connected in parallel with diodes. This ensures that the supply voltage across the logic is $V_{\rm DD} - 2V_D$, where V_D is the forward-biased voltage drop of a diode.

In [13], the authors describe an algorithm to optimally select low or high threshold implementations for each gate in the design. In [12], the authors address the problem of finding the best vector to utilize when the circuit is in standby mode. A genetic algorithm based solution technique is described for this problem.

Our approach avoids the use of additional circuitry to modify gate or bulk voltages in standby, and utilizes a dual threshold approach. However, unlike MTCMOS, the leakage of a design in our approach can be accurately estimated, and for large designs, it is always lower than the worst case MTCMOS leakage.

A shortened version of this manuscript appears in [17]. This manuscript significantly augments the results presented in [17].

III. Our Approach

This paper deals with low-leakage ASIC design using specialized standard cells. Based on the discussion of the previous section, we know that $I_{\rm ds}$ would be significantly larger when $V_{\rm ds} \gg nv_t$. This is because V_T drops due to DIBL when $V_{\rm ds}$ is large. This causes the first term of (1) to increase exponentially, while the parenthesized term of (1) is close to 1.

Our approach to leakage reduction attempts to ensure that the supply voltage is applied across more than one *turned-off* device and one of those devices is a high-V_T device. This is achieved by selectively introducing a high-V_T pMOS or nMOS supply gating device. By this design choice, we obtain standard cells with both *low and predictable* standby leakage currents.

Our goal is to design standard cells with *predictably* low leakage currents. To achieve this purpose, we design two variants of each standard cell. The two variants of each standard cell are designated "H" and "L." If the inputs of a cell during the *standby* mode of operation are such that the output has a high value, we minimize the leakage in the pull-down network. So a *footer* device (a high- V_T nMOS with its gate connected to $\overline{standby}$) is used. We call such a cell the "H" variant of the standard cell. Similarly, if the inputs of a cell during the standby mode of operation are such that the output has a low value, we minimize the leakage in the pull-up network by adding a *header* device (a high-V_T pMOS with its gate connected *standby*), and call such a cell the "L" variant of the standard cell.

This exercise, when carried out for a NAND3 gate, yields circuits shown in Fig. 1. Note that the MTCMOS circuit is also shown here. Although the pMOS and nMOS supply gating devices (equivalently called *header* and *footer* devices (devices shown shaded in Fig. 1) are shown in the circuit for the MTCMOS design, such devices are in practice shared by all the standard cells of a larger circuit block.

In our design approach, we utilized the same base standard cell library for all design styles. Our standard cell library

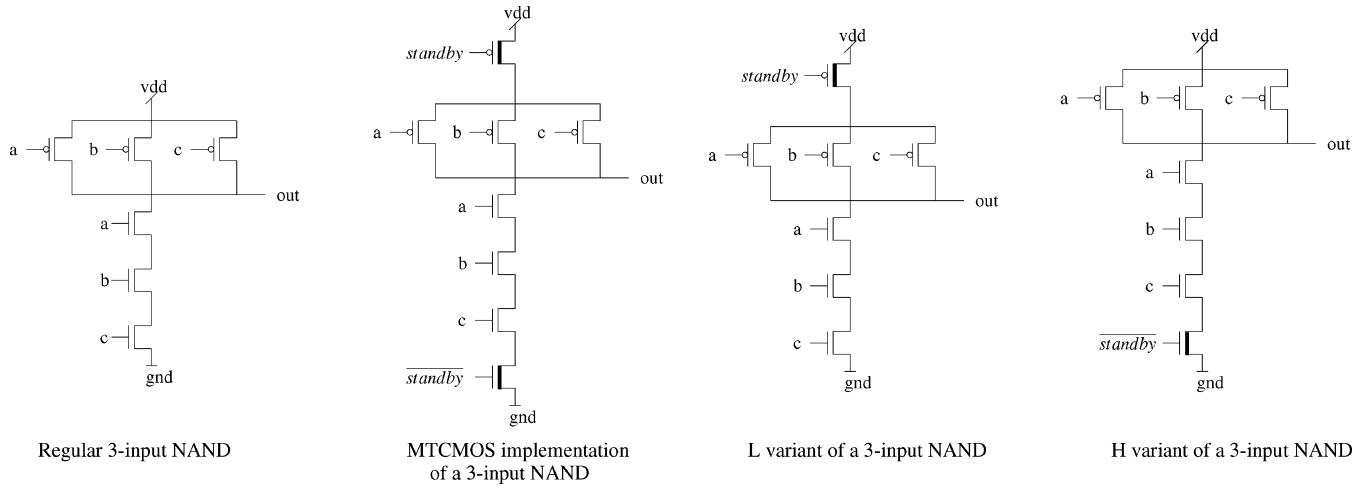


Fig. 1. Transistor level description (NAND3 gate).

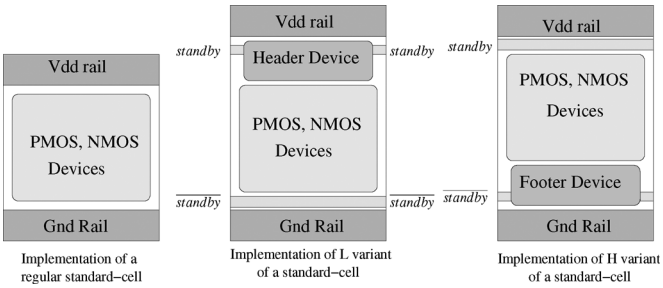


Fig. 2. Layout floor-plan of HL gates.

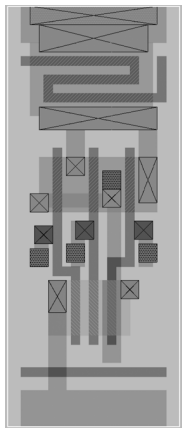


Fig. 3. Layout of NAND3-L cell.

consisted of INVA, INVB, NAND2A, NAND2B, NAND3, NAND4, NOR2, NOR3, NOR4, AND2, AND3, AND4, OR2, OR3, OR4, AOI21, AOI22, OAI21, and OAI22 cells. We utilized the *bsim100* predictive 0.1-μm model cards [18]. The devices have $V_T^N = 0.26$ V and $V_T^P = -0.30$ V. The header and footer devices we utilized had $V_T^N = 0.46$ V and $V_T^P = -0.50$ V. We sized the header and footer devices so that the worst-case output delay penalty over all gate input transitions was no larger than 15% as compared to the regular standard cell using low V_T transistors. In [5] too the power supply gating transistors were sized such that their simulated delay penalties were no larger than 15%. Additionally, if the delay penalty desired is less than 15%, then the gate area overheads are quite significant. The sizes of the devices of the regular standard cell were left unchanged in our MTCMOS and H/L cell variants.

If we were to modify the sizes of *all* devices (not just the header/footer devices), we anticipate that our cell area overheads would be much smaller, and the cells could be faster for a given area overhead. However, this would involve layout of H/L cells from scratch. For the results reported in this paper, we have made a decision to not modify the device sizes of the regular design in order to produce an approach which is easy to adopt. With this choice, we have been able to generate the layouts of the H/L standard cells by minimally modifying the layouts of the existing standard cells.

Our H/L cell layouts are derived from the existing standard cells by simply placing the $V_{\rm DD}$ and $G_{\rm ND}$ rails of a cell further apart, in order to introduce just enough additional space to insert the header/footer devices. This is shown schematically in Fig. 2. Note that in the H and L variants of the regular standard cell, the layout of the regular standard cell devices (the region labeled "pMOS, nMOS Devices") is not modified. The standby and $\overline{\rm standby}$ signals are routed by abutment, and run across the width of each H/L standard cell. The header and footer transistors are implemented in a space-efficient zig-zag configuration as shown in the layout of Fig. 3. This also allows the header and footer device regions to be available for over-the-cell routing. In our simulations we assumed that the width of the header and footer transistors to be equal to the center-line length of the poly shape. This is a common approximation used in circuit design. However for additional accuracy one can conceivably run existing commercial extraction tools to obtain an adjustment factor to account for the U-turns made in the poly shape. However, the adjustment factor is expected to be close to unity since there are only two U-turns in each H and L cell. Finally, our HL cells have more pin landing sites, to enable ease of routing. In this manner, we were able to design H/L layout variants of each cell in an area-efficient manner.

A. Design Methodology

In this paper, the overall design flow to implement a circuit using H/L standard cells is very similar to a traditional standard cell based design methodology. We first perform traditional mapping using regular standard cells. After determining a set of primary input assignments for the standby mode of operation, we simulate the circuit with these assignments to determine the output of each gate. If the output of a gate is high, we replace it with the corresponding H cell, and if it is low, with the L variant of the cell. Hence, the decision of which cell variant to utilize for any given circuit can be made in time linear in the size of the circuit.

The schemes discussed in [14]–[16] are similar to ours, but their authors do not mention that the leakage current in such a scheme is predictable. Also, in our HL methodology, the power supply gating devices are included within the standard cell itself for simplicity. This ensures that we do not have to use ungated additional power supply rails which are required in the schemes of [14]–[16]. We also perform detailed analysis of the delay-area tradeoffs for an extensive set of benchmark circuits, which is discussed in Section IV.

The determination of an optimal primary input assignment to utilize for the standby mode is actually a complex one. An algebraic decision diagram (ADD) [19] based framework can be used to solve this problem. In this approach, we would construct a decision diagram of a circuit topologically from primary inputs to primary outputs and assign each input vector a value of leakage based on the circuit state implied by that vector. Since the leakage current values for a cell differ for every input vector, we could potentially have an exponential number of decision nodes in the ADD. In order to develop an efficient algorithm for finding the best input vector, these different leakage values would need to be discretized. The granularity of this discretization would affect the optimality of the solution and the efficiency of the algorithm. Additionally, we could use a method of bounding the ADD leaf node values once a solution has been determined. For a scan enabled design, these primary inputs can be easily applied. If this is not the case, a phase-forcing circuit as discussed in [16] can be used to apply the required inputs to a combinational block.

B. Advantages and Disadvantages of Our Approach

The advantages of our methodology are as follows.

- By ensuring that each cell has a full-rail output value during standby operation, we make sure that the leakage of each standard cell, and therefore the leakage of a standard cell based design, are *precisely predictable*. Therefore, our methodology avoids the unpredictability of leakage that results when using the MTCMOS style of design. This unpredictability occurs due to the fact that in MTCMOS, cell outputs, inputs, and bulk voltages float to unknown values which are dependent on various processing and design factors.
- Since our inverting H/L cells utilize exactly one supply gating device (as opposed to two devices for MTCMOS), our cells exhibit better delay characteristics than MTCMOS for one output transition (the falling transition for L gates and rising transition for H gates). Though the authors of [5] mention that it possible to use only footer devices, their implementation uses both header and footer devices. Though using only a footer device will reduce the delay penalties, the leakage current increases as we show in Section IV.
- For MTCMOS designs, memory elements would require clean power and ground supplies if they were to retain state during standby mode [5]. With the HL approach, the inputs to a combinational block are fixed in the standby mode. Hence, the states of the memory elements that drive these inputs are also fixed. Therefore our technique can be applied to sequential elements as well (use header devices when the leakage path is through pMOS stack and use footer devices when the leakage path is through nMOS stack). Alternatively, we could utilize the same flip-flop design as in [5]. Either way, the HL approach would *not* require special clean supplies to be routed to the flip-flop cell, resulting in lower area utilization for sequential designs.
- For many of the standard cells, and particularly for larger cells which exhibit large values of leakage, our H/L cells exhibit much lower leakage current. However, there are cells for which our cells exhibit comparable or greater leakage than MTCMOS as well. This is quantified in Section IV.
- By implementing the header and footer devices in a layout-efficient manner, we ensure that the layout overhead of H/L standard cells is minimized. Our choice of layout also allows the header and footer device regions to be free for over-the-cell routing.

The disadvantages of our approach are as follows.

- The determination of the primary input assignments to utilize for the standby mode is a complex once. Although our current implementation makes this decision arbitrarily, it can be improved by applying the ideas described in Section III-A.
- Using the HL approach requires that the primary inputs to the circuit be driven to known values in the *standby* state. However if we assume that a combinational block of logic implemented using our approach is driven by flops with *scan*, then the required input vector can be simply scanned in before going into the *standby* state. Alternatively, special circuitry (such as a NAND2 or a NOR2 gate with the standby signal as one of the inputs) could be added at the primary inputs
- Though not done for the experiments in this paper, technology mapping tools need minor modifications in our methodology. Suppose we use a dynamic programming based technology mapper, the mapper would need to store the best match at any node as well as the logic state of that best match. For any new node that is being mapped, its logic state can therefore be determined, and so we would know whether to use a H or L cell for that mapping. In either case, we would know what delay or area value to use for an optimum match at that node. In reality, the problems of technology mapping and the determination of an optimal primary input vector are coupled.
- Our method requires that the standby signals be routed to each cell. However, we have shown a method to perform this efficiently, by designing the layout of H/L cells

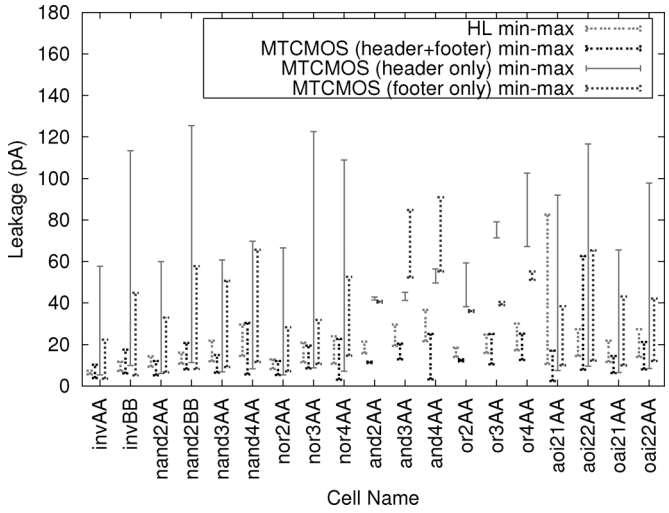


Fig. 4. Plot of leakage range of HL versus MT method.

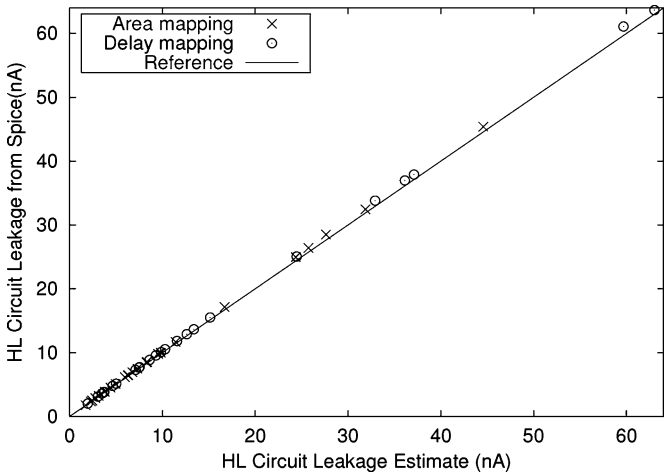


Fig. 5. Leakage of HL-SPICE versus HL method over circuits.

such that the routing is performed by abutment, while also leaving free space for over-the-cell routing above the region where the standby signals are run.

IV. Experimental Results

The standards cells we used were taken from the low-power standard cell library of [20]. Our standard cell library consisted of the following cells: INVA, INVB, NAND2A, NAND2B, NAND3, NAND4, NOR2, NOR3, NOR4, AND2, AND3, AND4, OR2, OR3, OR4, AOI21, AOI22, OAI21, and OAI22. The H and L variants of each of the standard cells were created by modifying (adding high- V_T header and/or footer devices as required) the regular cells. The header and footer devices used in the HL variants and for MTCMOS were sized such that the worst-case delays were within 15% of the regular standard cell worst-case delays. The sizes of the other transistors were not changed for reasons mentioned in Section III.

We used SPICE3f5 [21] for simulations of the standard cells. The nMOS and pMOS model cards used were derived from the *bsim100* model cards [18]. The threshold voltages of the high-V_T transistors were 200 mV greater than those of the regular devices. A supply voltage of 1.2 V was assumed.

After performing the design, layout and characterization of individual cells, we compared the leakage, delay and area characteristics of the HL, MTCMOS, and regular standard cell-based design methodologies for a set of circuits taken from the MCNC91 benchmark suite.

In Fig. 4, we plot the range of leakage values for each MTCMOS cell against the range of leakage values obtained using the corresponding HL cell. For the HL cells, all possible input vectors were applied for each cell. This gave us the range of leakage values possible for the HL cells. Finding the range of leakage for the MTCMOS cells, is not as straightforward as finding the leakage for HL cells since the inputs to the MTCMOS cell are not full-rail values during standby. For our experiments, we applied all possible voltage values from 0 to 1.2 V, in steps of 0.2 V, at each input of the MTCMOS cells and then found the minimum and maximum leakage currents. Note that in the Fig. 4, we have also compared the range of leakage values for MTCMOS cells using only header sleep transistors and for MTCMOS cells using only footer sleep transistors. From Fig. 4, we find that the range of leakage values for the MTCMOS cells using both header and footer sleep transistors is much smaller than the range of leakage values when only one of the devices is used. Hence, from this point on, we use only the MTCMOS cells with both header and footer devices for comparisons with our H/L cells.

A. Comparison of Placed and Routed Circuits

A set of circuits from the MCNC91 benchmarks were implemented using all three design methodologies (regular standard-cell, HL, and MTCMOS). Logic optimization and mapping were performed in the SIS [22] environment. The resulting leakage, area, and delay numbers were compared. For circuits designed using H/L type cells, each primary input signal was assumed to be logic low in standby mode. The choice of selecting the H or L variant for each standard cell was made as described in Section III-A.

1) Leakage Comparison: We first computed the leakage of each H/L cell based on the values of cell inputs implied by the primary input combination. Using this information, the leakage of the circuit mapped using the H/L gates was estimated by adding the leakage of the individual gates used. This is possible since the inputs to each gate in standby mode are known. We also ran SPICE on the mapped design, using the same primary input vector, to obtain a more accurate leakage estimate for the design. Fig. 5 is a scatter plot of the leakage values thus obtained, for all the circuits under consideration. From Fig. 5, we observe that for all the examples, the estimated leakage for the HL design and actual leakage obtained from SPICE are in very close agreement. This forms the basis for our claim that the *leakage for an HL design is precisely estimable* from the leakage values of each of its constituent gates. Thus, if one were to design low-leakage circuitry using the HL methodology, the standby power consumption can be computed with great accuracy. This is in stark contrast with MTCMOS based designs.

For the MTCMOS methodology, we determined the sum of the maximum and minimum leakage values of individual gates

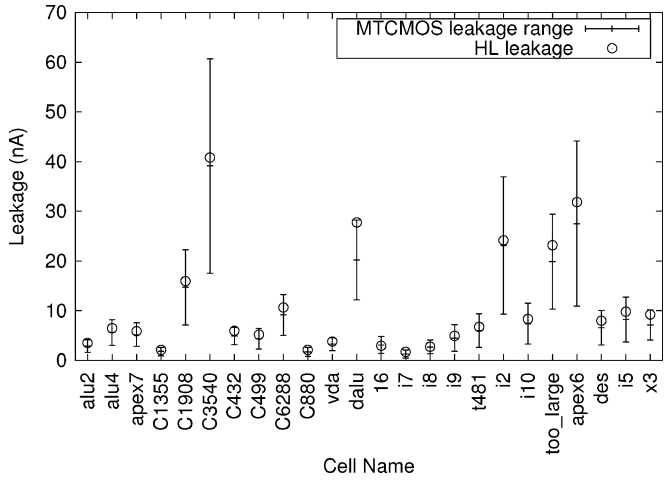


Fig. 6. Leakage of HL versus MT (circuits mapped for minimum area).

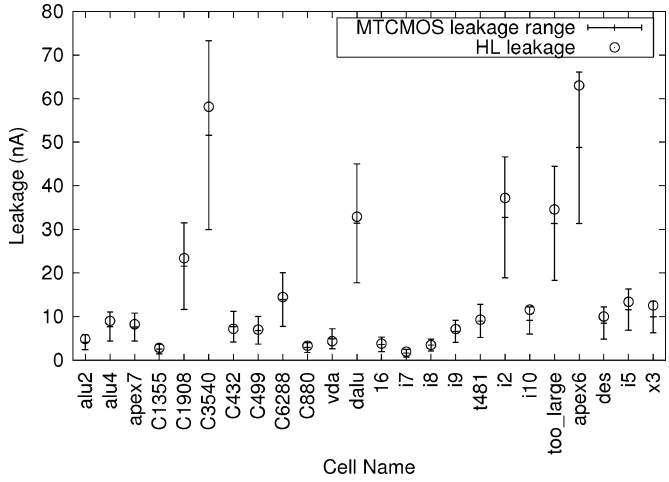


Fig. 7. Leakage of HL versus MT (circuits mapped for minimum delay).

(these values were also previously estimated from SPICE simulations and reported in Fig. 4). The results are presented in Figs. 6 and 7, and compared with the leakage of the HL methodology. In Fig. 6, the circuits were mapped for minimum area, while in Fig. 7, the circuits were mapped for minimum delay. In a mapped design, the inputs to the MTCMOS gates of the circuit would float in standby mode. Therefore, the precise leakage value for the MTCMOS design is unpredictable, hence we used the maximum and minimum values of MTCMOS leakage as mentioned in the description for Fig. 4. In practice, the actual value of the leakage current for a MTCMOS circuit may well be greater than the maximum value as computed above, based on the voltage values of gate inputs and bulk nodes.

Figs. 6 and 7 indicate that the leakage of a design implemented using HL cells can be much smaller than the maximum leakage of a MTCMOS design.

2) Delay Comparison: To compare the delay of the three techniques, we performed Exact Timing Analysis [23]. Given a mapped circuit, exact timing analysis returns the largest *sensitizable* delay for that circuit. As opposed to static timing analysis, exact timing eliminates false paths. We used the implementation

TABLE I
DELAY (ps) COMPARISON FOR ALL METHODS (DELAY MAPPING)

Example	Reg Delay	HL Delay	HL ovh.	MT Delay	MT ovh.
alu2	4146.65	4296.20	3.61	4546.15	9.63
alu4	5024.59	5135.15	2.20	5583.55	11.12
apex6	1660.15	1644.10	-0.97	1754.70	5.70
apex7	1959.00	1916.60	-2.16	2108.40	7.63
dalu	9270.03	10314.05	11.26	10494.15	13.21
des	14571.29	16690.05	14.54	16704.20	14.64
C1355	2567.91	2738.10	6.63	2922.80	13.82
C1908	3056.04	3403.45	11.37	3467.75	13.47
C3540	5756.18	6577.75	14.27	6537.05	13.57
C432	5309.39	5679.95	6.98	6015.25	13.29
C499	2289.99	2439.05	6.51	2586.20	12.93
C6288	13632.70	15528.65	13.91	15742.70	15.48
C880	2509.65	2853.90	13.72	2890.80	15.19
i2	610.55	652.70	6.90	665.95	9.07
i5	1136.75	1225.45	7.80	1232.35	8.41
i6	6698.08	7598.70	13.45	7610.40	13.62
i7	8074.18	9162.45	13.48	9174.15	13.62
i8	19027.58	21498.20	12.98	21799.45	14.57
i9	7370.84	8475.55	14.99	8503.00	15.36
i10	8479.30	8850.95	4.38	9680.85	14.17
t481	10040.29	11398.90	13.53	11374.05	13.28
too_large	4407.89	4809.00	9.10	4998.65	13.40
vda	3890.79	4329.05	11.26	4439.20	14.10
x3	2363.04	2653.60	12.30	2680.30	13.43
AVG			9.25%		12.61%

TABLE II
DELAY (ps) COMPARISON FOR ALL METHODS (AREA MAPPING)

Ckt.	Reg. Delay	HL Delay	HL ovh.	MT Delay	MT ovh
alu2	3971.00	4285.60	7.92	4474.70	12.68
alu4	6068.20	6797.55	12.02	6909.25	13.86
apex6	2248.85	2530.45	12.52	2500.20	11.18
apex7	1871.10	1925.60	2.91	2037.95	8.92
dalu	11868.45	12807.75	7.91	13198.00	11.20
des	19564.60	20593.90	5.26	22228.00	13.61
C1355	2952.80	3232.40	9.47	3383.60	14.59
C1908	4087.80	4689.80	14.73	4676.70	14.41
C3540	5730.85	6258.55	9.21	6528.40	13.92
C432	5220.30	5638.00	8.00	5893.10	12.89
C499	2723.60	3053.90	12.13	3117.60	14.47
C6288	11352.30	12912.65	13.74	13151.30	15.85
C880	2685.50	2963.30	10.34	2995.70	11.55
i2	703.00	763.60	8.62	787.60	12.03
i5	1154.70	1287.30	11.48	1270.80	10.05
i6	9182.30	10564.60	15.05	10409.20	13.36
i7	10549.85	11944.90	13.22	11781.10	11.67
i8	24974.05	28940.35	15.88	28675.30	14.82
i9	14746.35	16497.85	11.88	16576.30	12.41
i10	10335.00	11532.15	11.58	11664.95	12.87
t481	17192.70	19317.20	12.36	19092.50	11.05
too_large	4205.35	4650.85	10.59	4647.90	10.52
vda	5465.45	6140.05	12.34	6170.55	12.90
x3	3591.25	3986.60	11.01	3915.80	9.04
AVG			10.84%		12.49%

of exact timing (the *sense* package which is implemented in SIS [22]) from the authors of [23].

To run *sense*, we generated a modified library description file for each of the three techniques. This file, in SIS's *genlib* format, describes the rising and falling delay from each input pin to the output pin for all gates in the library. Each such delay is a tuple consisting of a constant delay and a load-dependent term. A standard cell library characterization script was utilized to automatically generate this *genlib* file for all three design styles.

The results of *sense* are described in Table I (for the case where mapping is done for delay minimization) and Table II

TABLE III
AREA (μ^2) COMPARISON FOR ALL METHODS (DELAY MAPPING)

Ckt.	2-Layer						3-Layer						4-Layer					
	Reg. Area	HL Area	HL ovh.	MT Area	MT ovh.	HL-MTovh.	Reg. Area	HL Area	HL ovh.	MT Area	MT ovh.	HL-MTovh.	Reg. Area	HL Area	HL ovh.	MT Area	MT ovh.	HL-MTovh.
alu2	2480.04	3203.56	29.17	3422.48	38.00	-6.40	1713.96	2560.36	49.38	2656.40	54.99	-3.62	1713.96	2560.36	49.38	2656.40	54.99	-3.62
alu4	5184.00	6400.00	23.46	6964.54	34.35	-8.11	3249.00	4542.76	39.82	5029.54	54.80	-9.68	3576.04	4542.76	27.03	5356.58	49.79	-15.19
apex6	4928.04	5565.16	12.93	6740.40	36.78	-17.44	3624.04	4542.76	25.35	5436.40	50.01	-16.44	4070.44	4542.76	11.60	5882.80	44.52	-22.78
apex7	1156.00	1600.00	38.41	1756.09	51.91	-8.89	1089.00	1444.00	32.60	1689.09	55.10	-14.51	1089.00	1459.24	34.00	1689.09	55.10	-13.61
dalu	10816.00	14352.04	32.69	15509.27	43.39	-7.46	9101.16	12678.76	39.31	13794.43	51.57	-8.09	9101.16	12678.76	39.31	13794.43	51.57	-8.09
des	46397.16	51710.76	11.45	56678.33	22.16	-8.76	50086.44	29790.76	-40.52	60367.61	20.53	-50.65	48664.36	28425.96	-41.59	58945.53	21.13	-51.78
C1355	3203.56	4542.76	41.80	5059.68	57.94	-10.22	4070.44	4542.76	11.60	5926.56	45.60	-23.35	3672.36	4542.76	23.70	5528.48	50.54	-17.83
C1908	3387.24	4761.00	40.56	4912.76	45.04	-3.09	2851.56	3969.00	39.19	4377.08	53.50	-9.32	3249.00	3969.00	22.16	4774.52	46.95	-16.87
C3540	7744.00	10120.36	30.69	10871.04	40.38	-6.91	5806.44	8100.00	39.50	8933.48	53.85	-9.33	5806.44	7779.24	33.98	8933.48	53.85	-12.92
C432	1169.64	1747.24	49.38	1819.41	55.55	-3.97	1169.64	1681.00	43.72	1819.41	55.55	-7.61	1197.16	1681.00	40.42	1846.93	54.28	-8.98
C499	2134.44	3069.16	43.79	3252.29	52.37	-5.63	1936.00	2704.00	39.67	3053.85	57.74	-11.46	3624.04	2704.00	-25.39	4741.89	30.85	-42.98
C6288	13041.64	17476.84	34.01	21098.58	61.78	-17.17	11620.84	16952.04	45.88	19677.78	69.33	-13.85	11620.84	16952.04	45.88	19677.78	69.33	-13.85
C880	1814.76	2480.04	36.66	2586.02	42.50	-4.10	1444.00	2134.44	47.81	2215.26	53.41	-3.65	1428.84	2134.44	49.38	2200.10	53.98	-2.98
i2	1024.00	1398.76	36.60	1358.97	32.71	2.93	817.96	1142.44	39.67	1152.93	40.95	-0.91	817.96	1142.44	39.67	1152.93	40.95	-0.91
i5	1918.44	2560.36	33.46	2676.54	39.52	-4.34	3249.00	2134.44	-34.30	4007.10	23.33	-46.73	2916.00	2116.00	-27.43	3674.10	26.00	-42.41
i6	3576.04	4705.96	31.60	4999.98	39.82	-5.88	4070.44	3969.00	-2.49	5494.38	34.98	-27.76	4070.44	3969.00	-2.49	5494.38	34.98	-27.76
i7	6177.96	6115.24	-1.02	8054.29	30.37	-24.07	4070.44	5212.84	28.07	5946.77	46.10	-12.34	4070.44	5212.84	28.07	5946.77	46.10	-12.34
i8	20449.00	26830.44	31.21	27105.78	32.55	-1.02	21609.00	20449.00	-5.37	28265.78	30.81	-27.65	21609.00	20449.00	-5.37	28265.78	30.81	-27.65
i9	5184.00	6561.00	26.56	6824.72	31.65	-3.86	4435.56	5745.64	29.54	6076.28	36.99	-5.44	4019.56	5745.64	42.94	5660.28	40.82	1.51
i10	28968.04	30765.16	6.20	35796.49	23.57	-14.06	22560.04	18117.16	-19.69	29388.49	30.27	-38.35	24649.00	18117.16	-26.50	31477.45	27.70	-42.44
t481	24964.00	33489.00	34.15	33259.85	33.23	0.69	20334.76	29104.36	43.13	28630.61	40.80	1.65	20334.76	29104.36	43.13	28630.61	40.80	1.65
too_large	5685.16	7396.00	30.09	7456.22	31.15	-0.81	3769.96	5212.84	38.27	5541.02	46.98	-5.92	3769.96	5270.76	39.81	5541.02	46.98	-4.88
vda	7992.36	10000.00	25.12	10111.07	26.51	-1.10	4900.00	6822.76	39.24	7018.71	43.24	-2.79	4928.04	6822.76	38.45	7046.75	42.99	-3.18
x3	6304.36	7499.56	18.96	8285.65	31.43	-9.49	4489.00	5745.64	27.99	6470.29	44.14	-11.20	4928.04	5745.64	16.59	6909.33	40.20	-16.84
AVG			29.08		38.94	-7.05			24.89		45.61	-14.96			20.70		43.97	-16.95

TABLE IV
AREA (μ^2) COMPARISON FOR ALL METHODS (AREA MAPPING)

Ckt.	2-Layer						3-Layer						4-Layer					
	Reg. Area	HL Area	HL ovh.	MT Area	MT ovh.	HL-MTovh.	Reg. Area	HL Area	HL ovh.	MT Area	MT ovh.	HL-MTovh.	Reg. Area	HL Area	HL ovh.	MT Area	MT ovh.	HL-MTovh.
alu2	2097.64	2560.36	22.06	2626.41	25.21	-2.51	1398.76	1764.00	26.11	1927.53	37.80	-8.48	1296.00	1764.00	36.11	1824.77	40.80	-3.33
alu4	4356.00	5685.16	30.51	5343.56	22.67	6.39	2894.44	3481.00	20.27	3882.00	34.12	-10.33	2601.00	3528.36	35.65	3588.56	37.97	-1.68
apex6	3721.00	4435.56	19.20	4667.28	25.43	-4.96	4070.44	3136.00	-22.96	5016.72	23.25	-37.49	4542.76	3113.64	-31.46	5489.04	20.83	-43.28
apex7	912.04	1296.00	42.10	1257.38	37.86	3.07	795.24	1142.44	43.66	1140.58	43.43	0.16	795.24	1142.44	43.66	1140.58	43.43	0.16
C1355	2323.24	3433.96	47.81	3324.24	43.09	3.30	2894.44	2981.16	3.00	3895.44	34.58	-23.47	2209.00	2981.16	34.96	3210.00	45.31	-7.13
C1908	2601.00	3624.04	39.33	3417.87	31.41	6.03	1918.44	2601.00	35.58	2735.31	42.58	-4.91	2894.44	2601.00	-10.14	3711.31	28.22	-29.92
C3540	6241.00	8281.00	32.69	7844.76	25.70	5.56	4225.00	5745.64	35.99	5828.76	37.96	-1.43	4489.00	5745.64	27.99	6092.76	35.73	-5.70
C432	817.96	1156.00	41.33	1116.43	36.49	3.54	729.00	1011.24	38.72	1027.47	40.94	-1.58	729.00	1011.24	38.72	1027.47	40.94	-1.58
C499	1764.00	2480.04	40.59	2381.99	35.03	4.12	1521.00	2134.44	40.33	2138.99	40.63	-0.21	1521.00	2135.36	40.39	2138.99	40.63	-0.17
C6288	10774.44	15525.16	44.09	15035.06	39.54	3.26	10281.96	12100.00	17.68	14542.58	41.44	-16.80	9025.00	12056.04	33.58	13285.62	47.21	-9.25
C880	1369.00	1989.16	45.30	1859.69	35.84	6.96	1197.16	1648.36	37.69	1687.85	40.99	-2.34	1197.16	1648.36	37.69	1687.85	40.99	-2.34
dalu	9254.44	11793.96	27.44	11834.39	27.88	-0.34	6304.36	8353.96	32.51	8884.31	40.92	-5.97	6304.36	8353.96	32.51	8884.31	40.92	-5.97
des	45710.44	47089.00	3.02	51786.20	13.29	-9.07	51529.00	22801.00	-55.75	57604.76	11.79	-60.42	51892.84	22560.04	-56.53	57968.60	11.71	-61.08
i2	772.84	1142.44	47.82	1041.92	34.82	9.65	817.96	985.96	20.54	1087.04	32.90	-9.30	817.96	985.96	20.54	1087.04	32.90	-9.30
i5	1681.00	2246.76	33.66	2210.91	31.52	1.62	1197.16	1600.00	33.65	1727.07	44.26	-7.36	1197.16	1600.00	33.65	1727.07	44.26	-7.36
i6	3433.96	3069.16	-10.62	4172.76	21.51	-26.45	3624.04	2560.36	-29.35	4362.84	20.39	-41.31	4070.44	2560.36	-37.10	4809.24	18.15	-46.76
i7	4928.04	5184.00	5.19	5868.12	19.08	-11.66	4070.44	3203.56	-21.30	5010.52	23.10	-36.06	3624.04	3203.56	-11.60	4564.12	25.94	-29.81
i8	17902.44	19656.04	9.80	21382.48	19.44	-8.07	18988.84	12769.00	-32.76	22468.88	18.33	-43.17	18769.00	12588.84	-32.93	22249.04	18.54	-43.42
i9	4329.64	4928.04	13.82	5415.02	25.07	-8.99	4070.44	3969.00	-2.49	5155.82	26.67	-23.02	4070.44	3969.00	-2.49	5155.82	26.67	-23.02
i10	28968.04	29584.00	2.13	32519.46	12.26	-9.03	19656.04	14448.04	-26.50	23207.46	18.07	-37.74	21609.00	13409.64	-37.94	25160.42	16.43	-46.70
t481	20107.24	25027.24	24.47	24616.10	22.42	1.67	12321.00	17056.36	38.43	16829.86	36.59	1.35	12321.00	17056.36	38.43	16829.86	36.59	1.35
too_large	5155.24	6432.04	24.77	6232.95	20.91	3.19	3069.16	4096.00	33.46	4146.87	35.11	-1.23	3249.00	4019.56	23.72	4326.71	33.17	-7.10
vda	7022.44	8427.24	20.00	8139.24	15.90	3.54	4277.16	5387.56	25.96	5393.96	26.11	-0.12	4225.00	5329.00	26.13	5341.80	26.43	-0.24
x3	5041.00	6822.76	35.35	6400.07	26.96	6.60	4984.36	4542.76	-8.86	6343.43	27.27	-28.39	5929.00	4542.76	-23.38	7288.07	22.92	-37.67
AVG			26.74		27.06	-0.52			11.82		32.47	-16.65			10.84		32.36	-17.55

(for the case where mapping is done for area minimization). For our benchmark suite of 24 examples, HL mapping exhibits a delay overhead of about 10% while MTCMOS exhibits an area overhead of 12.5%, compared to the regular method. As discussed earlier, the delay of the HL circuit is lower on account of the fact that only one transition of each gate is degraded in the process of modifying a gate for reduced leakage in the H/L approach. We also find that in two cases (apex7 and apex6 in Table I), the HL circuit actually has a small delay decrease. This is due to the fact that while adding a footer sleep device worsens the falling transition, the rising transition actually improves slightly. This is because the additional footer sleep device makes the path to ground more resistive and hence speeds up the rising transition. Similarly, falling transitions are improved slightly when a header sleep device is used. Hence in rare cases it is possible that a critical path gets sped up due to the addition of sleep transistors.

3) Area Comparison: We optimized and mapped our benchmark designs (for both minimum area and minimum delay) using SIS [22]. The circuits were then placed and routed using the Silicon Ensemble [24] tool set from Cadence Design Systems. Placement and routing was performed for both regular standard cell and H/L cell based circuits, using 2, 3, and 4 metal routing layers. This gave us an accurate measure of the actual die area required to design circuits using these two methodologies. For the MTCMOS methodology, the header and footer "sleep" transistors are large devices which are shared by all the gates in a design. According to [4], one can exploit information about simultaneous transitions in a circuit to size sleep transistors efficiently. As aforementioned, this approach is not feasible for random logic circuits. Therefore, for MTCMOS circuits, we found the sum of the sizes of the MTCMOS headers and footers of the individual gates in the design. Based on this information, we estimated the layout area overhead of MTCMOS. This overhead was then added to the area of the circuit implemented using regular cells. In an MTCMOS design, additional area needs to be devoted for routing an extra pair of power rails (see Section III-B). This was neglected since our designs were combinational in nature. For sequential circuits, the MTCMOS overhead would, therefore, be higher. Tables III and IV describe the area comparison results. The former table is obtained when technology mapping was performed for minimum delay, and the latter for minimum area. The tables show the total area for a 0.1-μm process for regular standard cell, HL cell, and MTCMOS based circuits. The percentage area overhead for the HL and MTCMOS methods is also shown.

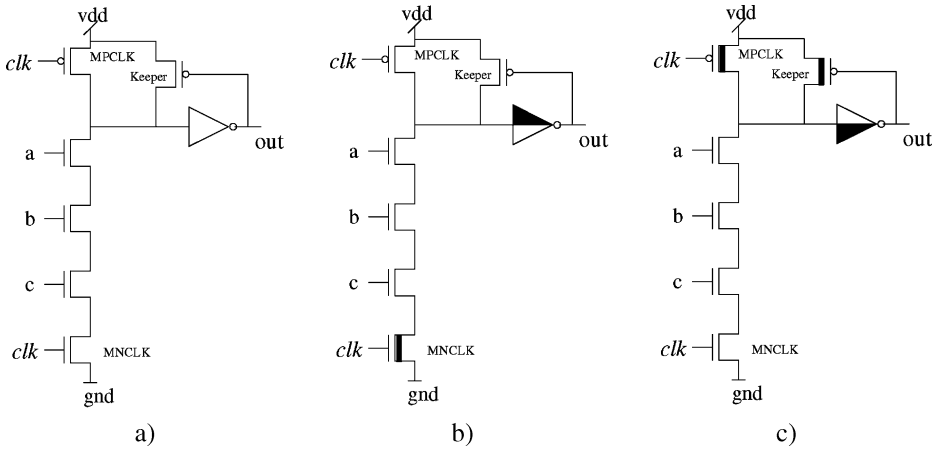


Fig. 8. Transistor level description (domino AND3 gate). (a) Regular implementation of a domino 3-input AND. (b) 3-input domino AND precharged in standby mode. (c) 3-input domino AND evaluated in standby mode.

We note that on average, the HL design methodology exhibits an 11%–30% area overhead compared to the regular design. However, the HL designs utilize on average up to 17% less area than the MTCMOS designs. As seen in Tables III and IV, the area overhead for MTCMOS does not decrease with increased metal layers, while the area overhead for HL does decrease. This is because the distributed nature of the sleep transistors in the HL scheme allows for more over-the-cell routing opportunities. The results validate that when more metal layers are used, the router can take advantage of over the cell routing and the area penalty for the HL methodology is reduced. For some examples, the HL designs exhibit a lower area than their regular counterparts. We conjecture that this is due to the fact that our HL cells are more router-friendly, with more over-the-cell routing space and also more pin landing sites.

V. Leakage Reduction in Domino Logic

Here, we explore how leakage power reduction is achieved in dynamic cells. Specifically we focus on domino logic cells due to their widespread popularity.

In standby mode, domino logic gates can either be in the *precharge* or *evaluate* state. In either case, if dual V_T technologies are used, devices which are turned *off* (devices in the cutoff mode of operation) in standby mode are implemented with high V_T. This can typically reduce leakage currents by about two orders of magnitude.

Fig. 8 illustrates the low leakage alternatives for a domino logic AND3 gate. Fig. 8(a) is a traditional domino AND3 gate. Fig. 8(b) illustrates the design of an AND3 domino logic gate which, in standby mode, is held in the precharge state (*clk* signal is logic-0). In this mode the pMOS pull-up device (MPCLK) is turned *on* and the nMOS pull-down stack is turned off. In the output inverter, the pMOS device is turned off and the nMOS device is turned on. The advantage in this method is that we have at least two devices turned *off* in series thus minimizing the leakage current. The footer device (MNCLK) and the pMOS device in the output inverter (illustrated by a dark triangle on the top part of the output inverter) are made high V_T to reduce leakage current further. However, both these devices are in the critical evaluate path of the domino logic gate, so the delay of the gate is increased when these devices are made high V_T. Therefore, these devices have to be up-sized to compensate for the increased delay. Rather than increasing the size of the footer device (MNCLK) alone, increasing the size of the rest of the devices in the nMOS stack results in smaller area penalties for the same delay.

Alternatively, the domino logic gate could be held in the *evaluate* state (nMOS stack turned *on*) during standby. In [4], the authors suggest such a method for a clock delayed domino logic scheme. An AND3 domino logic gate which is held in the *evaluate* state during standby is shown in Fig. 8(c). In standby mode the *clk* line is pulled high, thus turning off the pMOS pull-up device (MPCLK) and the nMOS in the output inverter. These devices are implemented with high V_T devices to keep the leakage current low. The *keeper* device is also made a high V_T device. The advantage of this scheme is that only the devices in the precharge path are made high V_T and any delay increase is exhibited only in that path. We found that the delay in the evaluate mode is in fact decreased slightly due to reduced leakage contention from the high V_T pMOS device (MPCLK).

A comparison of the leakages of the different schemes for a library of cells (cells compared were AND2, AND3, AND4, AND5, AND6, AOI21, AOI22, OAI21, OAI22, OR2, OR3, OR4, OR5, OR6, OR7, and OR8) is shown in Fig. 9. The scheme in which cells are held in *precharge* during standby is referred to as SP, and SE denotes the scheme in which cells are held in *evaluate* state during standby. In a regular domino logic gate, all devices are low V_T devices. Devices in the evaluate path of SP gates were up-sized such that the gate delay (in the evaluation phase) was made equal to the regular domino logic gate. As can be seen from Fig. 9, the leakage of SP and SE cells is dramatically lower (by about two orders of magnitude) than that of regular domino logic cells (for the same delay). Also it can be seen that the leakage for the SE scheme does not change much across the different gates. This is because the leaking devices, the pMOS pull-up device (MPCLK) and the nMOS device in the output inverter, are of the same size for all gates. Leakage for SE cells was determined to be lower than the SP cells, as illustrated in

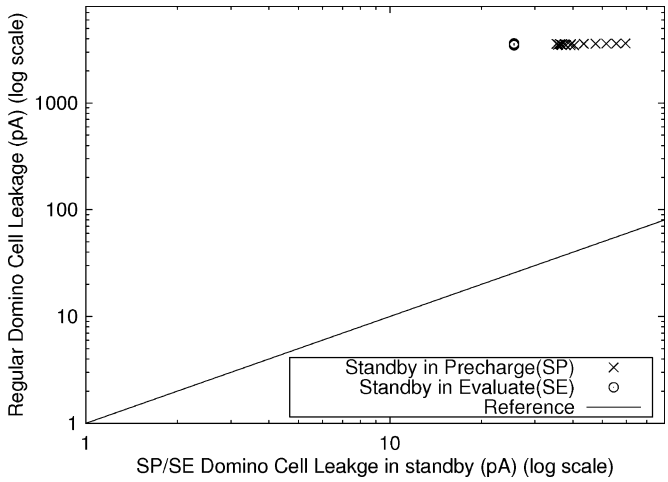
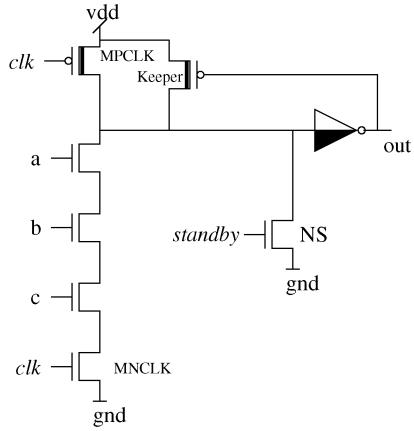


Fig. 9. Leakage of SE/SP versus regular domino cells.



3–input domino AND with with pull–down switch at the dynamic node

Fig. 10. Transistor level description of first SE domino gate in a chain.

TABLE V
LEAKAGE COMPARISON SE VERSUS SP

Ckt.	SP Leakage(pA)	SE Leakage(pA)	Ovh (%)
alu2	17516.82	12290.93	-29.83
alu4	36614.08	25913.37	-29.23
apex6	21543.77	15261.24	-29.16
apex7	7266.66	5146.83	-29.17
dalu	82461.74	58253.88	-29.36
des	166870.79	112001.09	-32.88
C1355	29497.98	21099.43	-28.47
C1908	27958.99	19588.67	-29.94
C3540	60278.10	41968.40	-30.38
C432	9184.09	6401.53	-30.30
C499	23250.06	16592.75	-28.63
C6288	165015.41	118914.73	-27.94
C880	14452.54	10140.02	-29.84
i2	3430.88	2048.49	-40.29
i5	7455.75	5095.62	-31.66
i6	10397.28	6913.66	-33.51
i7	12963.03	8501.24	-34.42
i8	52224.02	34542.67	-33.86
i9	16348.30	10626.55	-35.00
i10	101053.51	69443.76	-31.28
t481	47207.10	30164.03	-36.10
too_large	17053.89	11650.78	-31.68
vda	19747.06	12777.45	-35.29
x3	23492.55	16157.45	-31.22
Avg			-31.64

Fig. 9. This is because the high V_T devices in the SP cells had to be up-sized in order to avoid increased gate delays.

We also compared leakages of the SE and SP schemes for a set of circuits. The results are shown in Table V. The leakage for the SE scheme is on average 31% lower than the SP scheme.

From the above, it is clear that using SE domino logic gates is a better option from a delay, leakage and cell area standpoint (as compared to SP domino logic gates). For an SE domino logic gate, we need to ensure that all inputs of the gate are at logic-1 during standby mode. This can be done by gating the inputs of the first gate in a chain of domino logic cells. However, this will increase the delay of the gate during normal operation. The authors of [25] suggest a simple and elegant alternative. In this approach, an nMOS switch NS (as shown in Fig. 10) is used to pull down the dynamic node of the first gate in the chain. This switch is controlled by the standby signal. The only disadvantage of this method is that an additional standby signal is needed for the first gates in a chain of domino logic cells.

VI. CONCLUSION

In this paper, we have explored low-leakage standard cell based ASIC design methodologies for both static CMOS and domino logic. Our major contribution however is the development of a new methodology for low-leakage static CMOS designs, which we call the "HL" methodology. This "HL" methodology is based on ensuring that during standby operation, the supply voltage is applied across more than one *off* device and there is at least one *off* device in the leakage path which has a high V_T. For each standard cell in a library, we design two variants, the "H" and the "L" variant.

Our HL cells exhibit low leakage currents as do MTCMOS gates, but with the advantage that leakage currents in our methodology can be precisely estimated (unlike MTCMOS). We compared the two techniques using 24 placed-and-routed designs. We have shown that our methodology has a lower delay than MTCMOS, which is expected since our HL cells exhibit a delay degradation for only one output transition. Our HL designs exhibit *predictable* leakage values which are much lower than the maximum leakage for MTCMOS designs. Since leakage in MTCMOS designs is not precisely controllable, this is a significant improvement. Further, our HL designs exhibit an area overhead of approximately 21%–29% and 11%–27% over regular designs (for delay-optimal and area-optimal mapping respectively), and an area saving of up to 17% over MTCMOS designs.

The HL methodology utilizes existing mapping and place/route tools, and handles memory elements without additional routing overhead (unlike MTCMOS).

In the future, we plan to develop better algorithms to determine the best primary input vector to apply to a HL circuit during standby. Also, by redoing the layout of the H/L cells such that all devices of the cell are simultaneously optimized,

we expect that the delay as well as area characteristics of the HL design would improve.

With the downward scaling of $V_{\rm DD}$ in future technologies, the threshold voltages of both the high V_T and low V_T devices in the HL methodology will have to scale down too if the delays are to be kept within reasonable limits. However, this could increase the leakage current. So if leakage current is the overriding concern, the V_T of the high V_T power supply gating devices should not be scaled down. Though this may cause an increase in delays, this increase is in only one transition for each gate unlike traditional MTCMOS. Hence, the problems due to scaling of $V_{\rm DD}$ in future technologies are similar for both MTCMOS and HL methodologies, but are worse for MTCMOS.

REFERENCES

[1] ITRS, San Jose, CA, "The international technology roadmap for semiconductors," (2002). [Online]. Available: http://www.public.itrs.net/

[2] The Device Group, Dept. EECS, Univ. California, Berkeley, CA, "BSIM3 Homepage," (2001). [Online]. Available: http://www-device.eecs.berkeley.edu/~bsim3/intro.html

[3] J. Rabaey, *Digital Integrated Circuits: A Design Perspective*, ser. Prentice-Hall Electronics and VLSI Series. Englewood Cliffs, NJ: Prentice-Hall, 1996.

[4] J. T. Kao and A. P. Chandrakasan, "Dual-threshold voltage techniques for low-power digital circuits," *IEEE J. Solid-State Circuits*, vol. 35, no. 7, pp. 1009–1018, Jul. 2000.

[5] 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, Aug. 1995.

[6] T. Kuroda, T. Fujita, S. Mita, T. Nagamatsu, S. Yoshioka, K. Suzuki, F. Sano, M. Norishima, M. Murota, M. Kako, M. K. 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 J. Solid-State Circuits*, vol. 31, no. 11, pp. 1770–1779, Nov. 1996.

[7] H. Kawaguchi, K. Nose, and T. Sakurai, "A super cut-off CMOS (SCCMOS) scheme for 0.5-v supply voltage with picoampere stand-by current," *IEEE J. Solid-State Circuits*, vol. 35, no. 10, pp. 1498–1501, Oct. 2000.

[8] F. Assaderaghi, D. Sinitsky, S. A. Parke, J. Bokor, P. K. Ko, and C. Hu, "Dynamic threshold-voltage MOSFET (DTMOS) for ultra-low voltage VLSI," *IEEE Trans. Electron Devices*, vol. 44, no. 3, pp. 414–422, Mar. 1997.

[9] K. Kumagai, H. Iwaki, H. Yoshida, H. Suzuki, T. Yamada, and S. Kurosawa, "A novel powering-down scheme for low Vt CMOS circuits," in *Dig. Tech. Papers, Symp. VLSI Circuits*, 1998, pp. 44–45.

[10] I. Hyunsik, T. Inukai, H. Gomyo, T. Hiramoto, and T. Sakurai, "VTCMOS characteristics and its optimum conditions predicted by a compact analytical model," in *Proc. Int. Symp. Low Power Electron. Des.*, 2001, pp. 123–128.

[11] T. Inukai, T. Hiramoto, and T. Sakurai, "Variable threshold voltage CMOS (VTCMOS) in series connected circuits," in *Proc. Int. Symp. Low Power Electron. Design*, 2001, pp. 201–206.

[12] C. Zhanping, M. J. W. L. Wei, and W. Roy, "Estimation of standby leakage power in CMOS circuit considering accurate modeling of transistor stacks," in *Proc. Int. Symp. Low Power Electron. Des.*, 1998, pp. 239–244.

[13] Q. Wang and S. B. K. Vrudhula, "Static power optimization of deep submicron CMOS circuits for dual v_t technology," in *Dig. Tech. Papers, Int. Conf. Comput.-Aided Des. (ICCAD)*, 1998, pp. 490–496.

[14] 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, no. 11, pp. 1131–1135, Nov. 1993.

[15] D. Takashima, S. Watanabe, H. Nakano, Y. Oowaki, K. Ohuchi, and H. Tango, "Standby/active mode logic for sub-1-V operating ULSI memory," *IEEE J. Solid-State Circuits*, vol. 29, no. 4, pp. 441–447, Apr. 1994.

[16] K.-S. Min, H. Kawaguchi, and T. Sakurai, "Zigzag super cut-off CMOS (ZSCCMOS) block activation with self-adaptive voltage level controller: An alternative to clock-gating scheme in leakage dominant era," in *Dig. Tech. Papers, Int. Solid-State Circuits Conf. (ISSCC)*, 2003, pp. 400–502.

[17] N. Jayakumar and S. Khatri, "An ASIC design methodology with predictably low leakage, using leakage-immune standard cells," in *Proc., Int. Symp. Low Power Electron. Des.*, 2003, pp. 128–133.

[18] Y. Cao, T. Sato, D. Sylvester, M. Orshansky, and C. Hu, "New paradigm of predictive MOSFET and interconnect modeling for early circuit design," in *Proc. IEEE Custom Integr. Circuit Conf.*, 2000, pp. 201–204.

[19] R. I. Bahar, E. A. Frohm, C. M. Gaona, G. D. Hachtel, E. Macii, A. Pardo, and F. Somenzi, "Algebraic decision diagrams and their applications," *Formal Methods Syst. Des.*, vol. 10, no. 2/3, pp. 171–206, 1997.

[20] T. Burd, "CMOS standard cell 2_3lp library documentation," Univ. Calif. Berkeley, Berkeley, 1994.

[21] L. Nagel, "Spice: A computer program to simulate computer circuits," Univ. Calif., Berkeley, Tech. Rep. UCB/ERL Memo M520, 1995.

[22] E. M. Sentovich, K. J. Singh, L. Lavagno, C. Moon, R. Murgai, A. Saldanha, H. Savoj, P. R. Stephan, R. K. Brayton, and A. L. Sangiovanni-Vincentelli, "SIS: A system for sequential circuit synthesis," Electron. Res. Lab, Univ. Calif., Berkeley, Tech. Rep. UCB/ERL M92/41, 1992.

[23] P. C. McGeer, A. Saldanha, R. K. Brayton, and A. L. Sangiovanni-Vincetelli, "Logic synthesis and optimization," in *Delay Models and Exact Timing Analysis*. Boston, MA: Kluwer Academic, 1993, ch. 8.

[24] Cadence Design Systems, Inc., San Jose, CA, "Envisia silicon ensemble place-and-route reference," 1999.

[25] V. Kursun and E. G. Friedman, "Low swing dual threshold voltage domino logic," in *Proc. IEEE Great Lakes Symp. VLSI*, 2002, pp. 47–52.

Nikhil Jayakumar received the Bachelor's degree in electrical and electronics engineering from the University of Madras, Madras, India, and the Masters degree in electrical engineering from the University of Colorado at Boulder, Boulder. He is currently pursuing the Ph.D. degree in computer engineering from Texas A&M University, College Station.

During his graduate and doctoral studies, he has done research and published papers in many aspects of VLSI including formal verification, clock network design, routing, structured ASIC design, radiation-hard design, logic synthesis, LDPC decoder architectures, statistical timing, and low-power design. His current research interests focus on low-power design and more specifically on techniques to reduce, as well as exploit, leakage currents in VLSI.

Sunil P. Khatri received the B.Tech. degree in electrical engineering from the Indian Institute of Technology (IIT) Kanpur, India, the M.S. degree in electrical and computer engineering from the University of Texas, Austin, and the Ph.D. degree in electrical engineering and computer science from the University of California, Berkeley.

He was with Motorola, Inc., for four years, where he was a member of the design teams of the MC88110 and PowerPC 603 RISC microprocessors. He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, Texas A&M University, College Station. His research interests include logic synthesis, novel VLSI design approaches to address issues such as power, crosstalk, and cross-disciplinary applications of these topics. His research was supported by Intel Corporation, Lawrence Livermore National Laboratories, and the National Science Foundation. He has coauthored about 75 technical publications, five U.S. patent awards, one book, and a book chapter.

Dr. Khatri was a recipient of two Best Paper awards and two Best Paper Nominations.