An ASIC Design Methodology with Predictably Low Leakage, using Leakage-immune Standard Cells

Nikhil Jayakumar[‡] (jayakumn@colorado.edu)
Sunil P Khatri[‡] (spkhatri@colorado.edu)

[‡]Department of Electrical and Computer Engineering,
UCB 425, University of Colorado, Boulder CO 80309.

ABSTRACT

In this paper we introduce a low-leakage standard cell based ASIC design methodology which is based on the use of modified standard cells. These cells are designed to consume *extremely low and predictable* leakage currents in standby mode. For each cell in a standard cell library, we design two low-leakage variants of the cell. 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 vice versa. 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 designed and laid out our modified standard cells, and have performed experiments to compare placed-and-routed area, leakage and delays of our method against MTCMOS and a straightforward ASIC flow. Each design style we compare utilizes the same base standard cell library.

Our results show that designs obtained using our methodology have better speed and area characteristics than designs implemented in MTCMOS. The exact leakage current obtained for MTCMOS is highly unpredictable, while our method exhibits leakage currents which are precisely estimable. The leakage current for HL designs can be dramatically lower than the worst-case leakage of MTCMOS based designs, and two orders of magnitude compared to traditional standard cells. Also, a 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.

Categories and Subject Descriptors: B.7.1 [Integrated Circuits]: Types and Design Styles - Advanced technologies, Standard Cells, VLSI

General Terms: Design

Keywords: MTCMOS, leakage current, standby current, standard cells

1. INTRODUCTION

With diminishing process feature sizes and operating voltages, the control of (sub-threshold) leakage currents in modern VLSI designs is becoming a significant challenge. 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_{ds} of the device when the device is in the *cut-off* or *sub-threshold* region of operation. The expression for this current [1] is:

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

Here I_0 and V_{off}[1] are constants, while v_t is the thermal voltage (26mV at 300°K) and n is the sub-threshold swing parameter.

We note that I_{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. This is expected to be a major concern for VLSI design in the nanometer realm [2]. Therefore, effective control of sub-threshold I_{ds} is critical for the continued growth in portable electronic devices.

Another observation that can be made from equation 1 is that I_{ds} is significantly larger when $V_{ds} \gg nv_t$. For typical devices, this is satisfied when $V_{ds} \simeq VDD$. The reason for this is not only that the last term of equation 1 is close to unity, but also that with a large value of V_{ds}, V_T would be lowered due to drain induced barrier lowering[2] (DIBL) [3, 1]. 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, 5, 6, 7, 8, 9, 10, 11, 12, 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 2 discusses some previous work in this area. In Section 3 we describe our method to control leakage currents in an arbitrary standard cell. In Section 4 we present experimental results comparing our idea with MTCMOS and with traditional standard cell based ASIC design. Conclusions and future work are discussed in Section 5.

2. PREVIOUS WORK

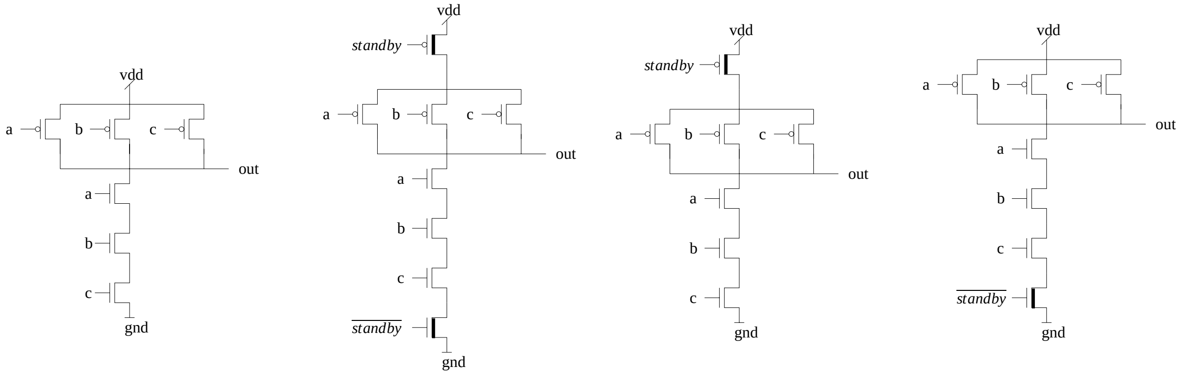
In recent times, leakage power reduction has received much attention in academic research as well as industrial applications. Several means of reducing leakage power have been proposed.

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 VDD is

ISLPED'03, August 25–27, 2003, Seoul, Korea.
Copyright 2003 ACM 1-58113-682-X/03/0008 ...$5.00.

[1] Typically $V_{off} = -0.08V$

[2] V_T decreases approximately linearly with increasing V_{ds}



Regular 3-input NAND

MTCMOS implementation of a 3-input NAND

L variant of a 3-input NAND

H variant of a 3-input NAND

Figure 1: Transistor Level Description (NAND3 gate)

lower than the diode turn-on voltage. Also, the increased capacitance of the gate signal slows the device down, which limits the use of this technique for partially depleted SOI designs.

More traditional design approaches have suggested the use of dual threshold devices [4] in an MTCMOS configuration[3]. 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 PLL using a 0.5μm 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, making MTCMOS less appealing. Since cell inputs and outputs as well as bulk nodes float in an MTCMOS design operating in standby mode, their voltages (which significantly affect the leakage of the design[4]) are determined by process and parasitic considerations and are extremely hard to determine or control. This results in a situation where MTCMOS designs can have a large range of leakage currents. 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 is proposed, in which the MTCMOS sleep devices are connected in parallel with diodes. This ensures that the supply voltage across the logic is $VDD - 2V_D$, where V_D is the forward-biased voltage drop of a diode.

Another methodology for controlling leakage is the Variable-threshold [6, 11, 10] (often called VTCMOS) approach. In such an approach, the device threshold voltages are controlled dynamically by modifying the device bulk voltage. This requires the design of complex control circuitry to control the bulk voltage. The authors of [6] implemented a Discrete Cosine Transform (DCT) core using this approach. Other approaches [7] over-drive the gate of a PMOS device which gates the VDD supply, thereby reducing the leakage dramatically. This again requires the design of complex circuitry to generate the special over-driven voltage values.

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 almost always lower than the worst case MTCMOS leakage.

3. OUR APPROACH

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

Our approach to leakage reduction attempts to ensure that the entire supply voltage is not applied across a single 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.

Although designs implemented using MTCMOS exhibit low leakage currents, the exact value of leakage of an MTCMOS design varies widely and is dependent on process and design factors. 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 all these nodes are floating in standby, precise prediction or control of leakage is impossible in MTCMOS. The voltage of these floating nodes can significantly affect the device threshold voltages. Since sub-threshold I_{ds} has an exponential dependence on threshold voltage as seen in equation 1, this situation is undesirable. Although the maximum value of leakage for an MTCMOS design is significantly lower than that of a design implemented using traditional standard cells, it would be desirable to design a leakage reduction approach with low *and predictable* 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. 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. 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, and call such a cell the "L" variant of the standard cell.

The minimization of leakage in the pull-down network (for a H cell) is achieved by gating the GND supply with a high V_T NMOS device connected to the $\overline{standby}$ signal. An analogous modification is made for L cells.

[3] MTCMOS utilizes NMOS and PMOS power supply gating devices

[4] Cell input and output voltages affect the leakage of a gate as seen in equation 1. Bulk voltage V_b affects V_T through body effect, and sub-threshold leakage has an exponential dependence of V_T as seen in equation 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 exercise, when carried out for a NAND3 gate, yields circuits shown in Figure 1. Note that the MTCMOS circuit is also shown here. Although the PMOS and NMOS supply gating devices (equivalently called *header* and *footer* devices[5] 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[6] for all design styles. We utilized the *bsim100* predictive 0.1μm model cards from [14]. The devices have $V_T^N = 0.26V$ and $V_T^P = -0.30V$. The header and footer devices we utilized had $V_T^N = 0.46V$ and $V_T^P = -0.50V$. 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. The sizes of the devices of the regular standard cell were left unchanged in our MTCMOS and H/L cell variants.

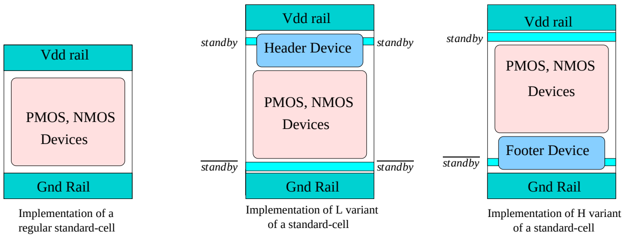


Figure 2: Layout Floorplan of HL gates

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 owing to time constraints[7]. 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 VDD and GND 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 Figure 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 $\overline{standby}$ 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 Figure 3. This also allows the header and footer device regions to be available for over-the-cell routing. 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 a area-efficient manner.

3.1 Design Methodology

The overall design flow to implement a circuit using H/L standard cells is very similar to a traditional standard cell based design. 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

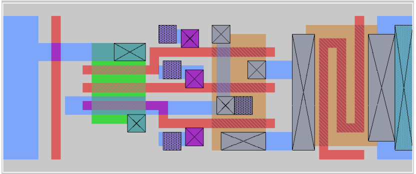


Figure 3: Layout of NAND3-L cell (rotated 90° clockwise)

gate is high, we replace it with the corresponding H cell, and vice versa. 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 determination of the optimal primary input assignments to utilize for the standby mode is actually a complex one. We plan to introduce an Algebraic Decision Diagram [15] based framework to determine the primary input vector which should be applied in standby operation. In such an approach, we would construct the decision diagram of a circuit topologically from primary inputs to primary outputs, assigning each input vector a value of leakage based on the circuit state implied by that vector. In the worst case, we would have an exponential number of decision nodes as leaves of the ADD, but by discretizing the leakage values at the leaves, we could reduce this complexity. It would be interesting to determine the tradeoff between the granularity of discretization and the accuracy of the resulting vector. Additionally, we plan to use a method of bounding the ADD leaf node values once a solution has been determined.

3.2 Advantages and Disadvantages of Our Approach

The advantages of our methodology are:

• 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 vice-versa). Because of this, the delay of circuits mapped using H/L cells is smaller than the corresponding delay for MTCMOS based designs, as we shall quantify in Section 4.

• For MTCMOS designs, memory elements would require clean power and ground supplies if they were to retain state during standby mode [5]. With the H/L approach however, we would utilize the same flip-flop design as in [5], but 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 4.

• 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. • As described earlier, technology mapping of a design using H/L cells can be easily performed without modifying existing tools.

The disadvantages of our approach are:

• The determination of the primary input assignments to utilize

[5] These devices are shown shaded in Figure 1

[6] Our standard cell library consisted of INVA, INVB, NAND2A, NAND2B, NAND3, NAND4, NOR2, NOR3, NOR4, AND2, AND3, AND4, OR2, OR3, OR4, AOI21, AOI22, OAI21 and OAI22 cells

[7] Preliminary SPICE simulations have demonstrated that recreating the standard cells in this manner results in a much lower delay and area penalty.

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 3.1.

• 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 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.

4. EXPERIMENTAL RESULTS

The standards cells we used were taken from the low-power standard cell library of [16]. 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 3.

We used SPICE3f5 [17] for simulations of the standard cells. The NMOS and PMOS model cards used were derived from the *bsim100* model cards [14]. The threshold voltages of the high-V_T transistors were 200mV greater than those of the regular devices.

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.

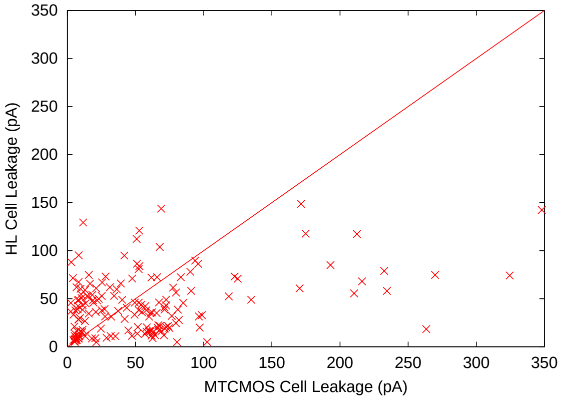


Figure 4: Leakage of HL versus MT method (over *all* cells)

Figure 4 is a scatter plot of leakage values of the HL and MTCMOS design approaches (derived for all input vectors and all gates). We observe that for several data points, the MTCMOS leakage is significantly more than the HL leakage, whereas for the remaining data points the two techniques have a roughly equal number of “wins”. This indicates that our HL cells have slightly better leakage characteristics than the MTCMOS cells. The major advantage of our cells over MTCMOS is the predictability of their leakage values.

Also, the leakage of MTCMOS and HL cells is dramatically lower than that of traditional cells as shown in Figure5 (on a double logarithmic scale). This is as expected, and indicates that both the MTCMOS and HL techniques exhibit significantly lower leakage than the regular cells.

In Figure 6, we plot the range of leakage values for each MTCMOS cell against the range of leakage values obtained using the corresponding HL cell. Note that for the MTCMOS cells with large maximum leakage, the HL cells exhibit a significantly smaller maximum leakage.

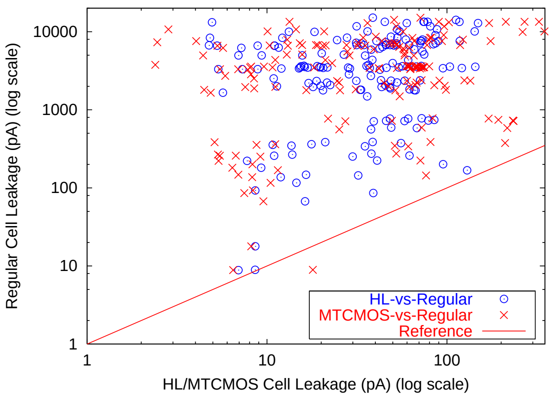


Figure 5: Leakage of HL/MT versus regular cells (over *all* cells)

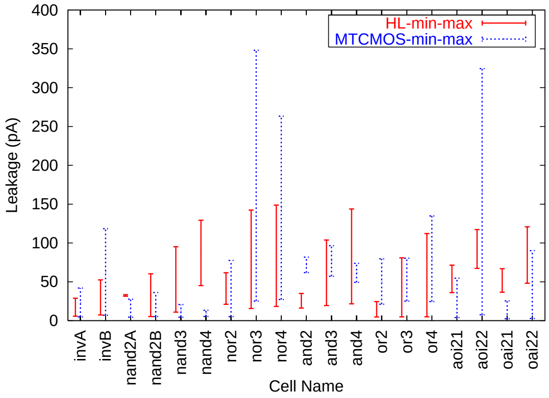


Figure 6: Plot of leakage range of HL versus MT method

4.1 Comparison of Placed and Routed Circuits

A set of circuits from the MCNC91 benchmarks were implemented using all three design methodologies. Logic optimization and mapping were performed using SIS [18]. 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 3.1.

4.1.1 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. Figure 7 is a scatter plot of the leakage values thus obtained, for all the circuits under consideration. From Figure 7, 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 a 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 (these values were also previously estimated from SPICE simulations and reported in Table 6). The results are presented in Figures 8 and 9,

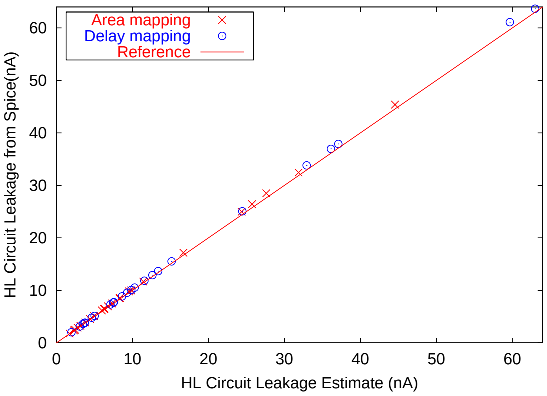


Figure 7: Leakage of HL-spice versus HL method over circuits

and compared with the leakage of the HL methodology. In Figure 8, the circuits were mapped for minimum area, while in Figure 9, the circuits were mapped for minimum delay. Note that for MTCMOS (as with all other) gates, SPICE simulations of individual gates were performed for all possible input vectors of the gate. However, 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 described above. 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.

Figures 8 and 9 indicate that the leakage of a design implemented using HL cells can be dramatically smaller than the maximum leakage of a MTCMOS design.

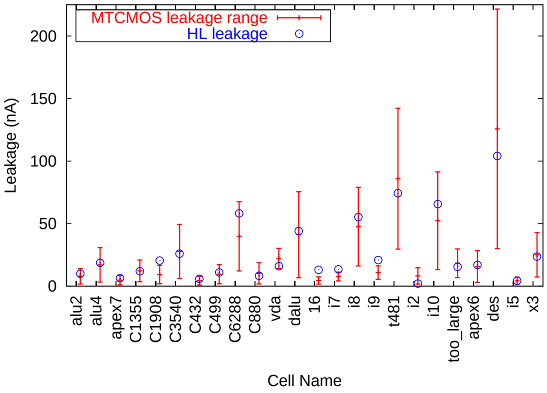


Figure 8: Leakage of HL versus MT (circuits mapped for min. area)

4.1.2 *Delay Comparison*

To compare the delay of the three techniques, we performed Exact Timing Analysis [19]. 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 of exact timing (the *sense* package which is implemented in SIS [18]) written by the authors of [19].

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.

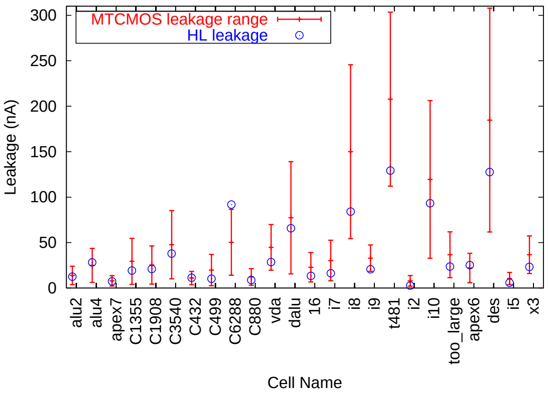


Figure 9: Leakage of HL versus MT (circuits mapped for min. delay)

The results of *sense* are described in Table 1 (for the case where mapping is done for delay minimization) and Table 2 (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.

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
apex7	1959.00	1916.60	-2.16	2108.40	7.63
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
vda	3890.79	4329.05	11.26	4439.20	14.10
dalu	9270.03	10314.05	11.26	10494.15	13.21
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
t481	10040.29	11398.90	13.53	11374.05	13.28
i2	610.55	652.70	6.90	665.95	9.07
i10	8479.30	8850.95	4.38	9680.85	14.17
too_large	4407.89	4809.00	9.10	4998.65	13.40
apex6	1660.15	1644.10	-0.97	1754.70	5.70
des	14571.29	16690.05	14.54	16704.20	14.64
i5	1136.75	1225.45	7.80	1232.35	8.41
x3	2363.04	2653.60	12.30	2680.30	13.43
AVG			9.25%		12.61%

Table 1: Delay (ps) Comparison for all Methods (delay mapping)

4.1.3 *Area Comparison*

We optimized and mapped our benchmark designs (for both minimum area and minimum delay) using SIS [18]. The circuits were then placed and routed using the Silicon Ensemble [20] tool set from Cadence Design Systems. Placement and routing was performed for both regular standard cell and H/L cell based circuits, using 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 we have stated earlier, this approach is infeasible 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

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
apex7	1871.10	1925.60	2.91	2037.95	8.92
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
vda	5465.45	6140.05	12.34	6170.55	12.90
dalu	11868.45	12807.75	7.91	13198.00	11.20
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
t481	17192.70	19317.20	12.36	19092.50	11.05
i2	703.00	763.60	8.62	787.60	12.03
i10	10335.00	11532.15	11.58	11664.95	12.87
too_large	4205.35	4650.85	10.59	4647.90	10.52
apex6	2248.85	2530.45	12.52	2500.20	11.18
des	19564.60	20593.90	5.26	22228.00	13.61
i5	1154.70	1287.30	11.48	1270.80	10.05
x3	3591.25	3986.60	11.01	3915.80	9.04
AVG			10.84%		12.49%

Table 2: Delay (ps) Comparison for all Methods (area mapping)

regular cells. In an MTCMOS design, additional area needs to be devoted for routing an extra pair of power rails (see section 3.2). This was neglected since our designs were combinational in nature. Tables 3 and 4 describe the area comparison results. The former table is obtained when technology mapping was performed for minimum delay, and the latter for minimum area. Column 4 lists the area overhead of the HL designs compared the the regular design. Column 6 represents the corresponding overhead of MTCMOS designs. Column 7 represents the overhead of the HL design compared to the MTCMOS design.

We note that on average, the HL design methodology exhibits a 11-21% area overhead compared to the regular design. However, the HL designs utilize on average 17% less area than the MTCMOS designs. 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.

Ckt.	Reg. Area	HL Area	HL ov	MT Area	MT ov	HL-MT ov
alu2	1713.96	2560.36	49.38	2656.40	54.99	-3.62
alu4	3576.04	4542.76	27.03	5356.58	49.79	-15.19
apex7	1089.00	1459.24	34.00	1689.09	55.10	-13.61
C1355	3672.36	4542.76	23.70	5528.48	50.54	-17.83
C1908	3249.00	3969.00	22.16	4774.52	46.95	-16.87
C3540	5806.44	7779.24	33.98	8933.48	53.85	-12.92
C432	1197.16	1681.00	40.42	1846.93	54.28	-8.98
C499	3624.04	2704.00	-25.39	4741.89	30.85	-42.98
C6288	11620.84	16952.04	45.88	19677.78	69.33	-13.85
C880	1428.84	2134.44	49.38	2200.10	53.98	-2.98
vda	4928.04	6822.76	38.45	7046.75	42.99	-3.18
dalu	9101.16	12678.76	39.31	13794.43	51.57	-8.09
i6	4070.44	3969.00	-2.49	5494.38	34.98	-27.76
i7	4070.44	5212.84	28.07	5946.77	46.10	-12.34
i8	21609.00	20449.00	-5.37	28265.78	30.81	-27.65
i9	4019.56	5745.64	42.94	5660.28	40.82	1.51
t481	20334.76	29104.36	43.13	28630.61	40.80	1.65
i2	817.96	1142.44	39.67	1152.93	40.95	-0.91
i10	24649.00	18117.16	-26.50	31477.45	27.70	-42.44
too_large	3769.96	5270.76	39.81	5541.02	46.98	-4.88
apex6	4070.44	4542.76	11.60	5882.80	44.52	-22.78
des	48664.36	28425.96	-41.59	58945.53	21.13	-51.78
i5	2916.00	2116.00	-27.43	3674.10	26.00	-42.41
x3	4928.04	5745.64	16.59	6909.33	40.20	-16.84
AVG			20.70		43.97	-16.95

Table 3: Area (μ^2) Comparison for all Methods (delay mapping)

5. CONCLUSIONS

In this paper, we have described a low-leakage standard cell based ASIC design methodology. This "HL" methodology is based on ensuring that during standby operation, the supply voltage is applied across more than one device. For each standard cell in a library, we design two variants, the "H" and the "L" variant.

Our H/L 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 H/L 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 11 or 21% over regular designs (for area-optimal or delay-optimal mapping respectively), and an area saving of approximately 17% over MTCMOS designs.

Ckt.	Reg. Area	HL Area	HL ov	MT Area	MT ov	HL-MT ov
alu2map	1296.00	1764.00	36.11	1824.77	40.80	-3.33
alu4map	2601.00	3528.36	35.65	3588.56	37.97	-1.68
apex7map	795.24	1142.44	43.66	1140.58	43.43	0.16
C1355map	2209.00	2981.16	34.96	3210.00	45.31	-7.13
C1908map	2894.44	2601.00	-10.14	3711.31	28.22	-29.92
C3540map	4489.00	5745.64	27.99	6092.76	35.73	-5.70
C432map	729.00	1011.24	38.72	1027.47	40.94	-1.58
C499map	1521.00	2135.36	40.39	2138.99	40.63	-0.17
C6288map	9025.00	12056.04	33.58	13285.62	47.21	-9.25
C880map	1197.16	1648.36	37.69	1687.85	40.99	-2.34
vdamap	4225.00	5329.00	26.13	5341.80	26.43	-0.24
dalumap	6304.36	8353.96	32.51	8884.31	40.92	-5.97
i6map	4070.44	2560.36	-37.10	4809.24	18.15	-46.76
i7map	3624.04	3203.56	-11.60	4564.12	25.94	-29.81
i8map	18769.00	12588.84	-32.93	22249.04	18.54	-43.42
i9map	4070.44	3969.00	-2.49	5155.82	26.67	-23.02
t481map	12321.00	17056.36	38.43	16829.86	36.59	1.35
i2map	817.96	985.96	20.54	1087.04	32.90	-9.30
i10map	21609.00	13409.64	-37.94	25160.42	16.43	-46.70
too_large	3249.00	4019.56	23.72	4326.71	33.17	-7.10
apex6map	4542.76	3113.64	-31.46	5489.04	20.83	-43.28
desmap	51892.84	22560.04	-56.53	57968.60	11.71	-61.08
i5map	1197.16	1600.00	33.65	1727.07	44.26	-7.36
x3map	5929.00	4542.76	-23.38	7288.07	22.92	-37.67
AVG			10.84		32.36	-17.55

Table 4: Area (μ^2) Comparison for all Methods (area mapping)

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 re-doing 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.

6. REFERENCES

[1] "BSIM3 Homepage." `http://www-device.eecs.berkeley.edu/~bsim3/arch_ftp.html`.

[2] "The International Technology Roadmap for Semiconductors." `http://public.itrs.net/`, 2002.

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

[4] J. T. Kao and A. P. Chandrakasan, "Dual-threshold voltage techniques for low-power digital circuits," *IEEE Journal of Solid-State Circuits*, vol. 35, 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 Journal of Solid-State Circuits*, vol. 30, 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 Journal of Solid-State Circuits*, vol. 31, 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 Journal of Solid-State Circuits*, vol. 35, 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 Transactions on Electron Devices*, vol. 44, 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 *Digest of Technical Papers, Symposium on VLSI circuits*, pp. 44–45, Jun 1998.

[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 *International Symposium on Low Power Electronics and Design*, pp. 123–128, 2001.

[11] T. Inukai, T. Hiramoto, and T. Sakurai, "Variable threshold voltage cmos (VTCMOS) in series connected circuits," in *International Symposium on Low Power Electronics and Design*, pp. 201–206, 2001.

[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 *International Symposium on Low Power Electronics and Design*, pp. 239–244, 1998.

[13] Q. Wang and S. B. K. Vrudhula, "Static power optimization of deep submicron CMOS circuits for dual v_t technology," in *Digest of Technical Papers, International Conference on Computer-Aided Design (ICCAD)*, pp. 490–496, Nov 1998.

[14] "BSIM3 Homepage." `http://www-device.eecs.berkeley.edu/~bsim3/intro.html`.

[15] 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 in Systems Design*, vol. 10, no. 2/3, pp. 171–206, 1997.

[16] T. Burd, *CMOS Standard Cell 2_3lp Library Documentation*. U C Berkeley, Mar 1994.

[17] L. Nagel, "Spice: A computer program to simulate computer circuits," in *University of California, Berkeley UCB/ERL Memo M520*, May 1995.

[18] 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," Tech. Rep. UCB/ERL M92/41, Electronics Research Lab, Univ. of California, Berkeley, CA 94720, May 1992.

[19] P. C. McGeer, A. Saldanha, and R. K. B. A. L. Sangiovanni-Vincetelli, *Delay models and exact timing analysis*, ch. 8. Logic Synthesis and Optimization, Kluwer Academic Publishers, 1993.

[20] Cadence Design Systems, Inc., 555 River Oaks Parkway, San Jose, CA 95134, USA, *Envisia Silicon Ensemble Place-and-route Reference*, Nov 1999.