Runtime Leakage Minimization through Probability-Aware Dual-Vt or Dual-Tox Assignment

Dongwoo Lee, David Blaauw, Dennis Sylvester
{dongwool,blaauw,dmcs}@umich.edu
University of Michigan, Ann Arbor, MI

Abstract - With process scaling runtime leakage current, when the circuit is operating, has become a major concern in addition to traditional standby mode leakage. In this paper we propose a new leakage reduction method that specifically targets runtime leakage current. We first observe that the state probabilities of nodes in a circuit tend to be skewed, meaning that they have either a high or a low value. We then propose a method that exploits these skewed state probabilities by setting only those transistors to high-V_t (thick-oxide) that have a high likelihood of being OFF (ON) and hence contributing significantly to the total runtime leakage. Accordingly, we also propose a library specifically tailored for the proposed approach, where V_t and T_{ox} assignment with favorably trade-offs under skewed input probabilities are provided. The optimization algorithm performs simultaneous sizing, V_t and T_{ox} assignment and shows substantial leakage improvement over probability-unaware optimization.

1 Introduction

In recent years, standby mode and runtime leakage power have become significant concerns as process dimensions and supply voltage continue to scale down. Up to 54% of the total power dissipation is projected to be leakage power dissipation at the 65nm node [1]. To address leakage current in standby mode, the MTCMOS approach was proposed where a high-V_t gating transistor is inserted in series with the power supply [2]. This method incurs routing overhead for virtual power supplies and requires special latches to preserve state in standby mode [3]. In a different approach, a dedicated sleep input vector that minimizes leakage current is assigned to a circuit in standby mode [4]. This approach uses modified flip-flops that force the output to the required state [5]. However, the leakage reduction is small - typically in the range of 10-30% [6].

The dual-V_t approach reduces leakage current by assigning transistor threshold voltages using a process where both high and low-V_t transistors are available. To reduce leakage current, non-critical gates in the circuit are assigned to high-V_t, while critical circuit portions are assigned to low-V_t [7][8][9]. This approach was extended for standby mode operation, by combining the V_t assignment with sleep state assignment using a branch and bound search method [10]. This method is based on the observation that, given a known input state for a gate, the subthreshold leakage current (I_{sub}) of that gate can be reduced by setting only OFF transistors on each path from V_{dd} to *Gnd* to high-V_t, because only OFF transistors are responsible for I_{sub}. Therefore, this approach improves the trade-off between leakage and performance compared to V_t assignment with unknown input states where most or all of the transistors must be set to high-V_t before a significant improvement in the leakage current is observed. However, while this approach significantly improves the leakage current in standby mode, it is not applicable to runtime leakage while the circuit is operating, as the circuit state is not known in this case.

Traditionally, runtime leakage power has been of less concern than standby mode leakage since in runtime dynamic power dissipation has been significantly greater than static power dissipation. Nevertheless, due to aggressive process scaling the leakage power dissipation is becoming comparable to dynamic power in high performance processor designs [11]. Therefore, new approaches for reducing leakage power in runtime mode are needed.

In this paper, we propose a new method to reduce leakage current in runtime mode. Our approach leverages dual-V_t / dual-T_{ox} technology and performs simultaneous circuit sizing, V_t assignment and T_{ox} assignment. However, in order to improve the leakage / performance trade-off, we exploit the *state probabilities* of nodes in the circuit during runtime combined with a specially tailored cell library that takes advantage of frequently occurring skewed gate input probabilities. The state probability of a node is the probability of that node being in a high state. Since leakage current depends strongly on the state of the gate inputs, leakage current in runtime mode is strongly influenced by the state probabilities of the nodes. For example, if an NMOS transistor has a very high state probability in runtime mode, it will be OFF and leaking only a small portion of time and its contribution to the overall leakage current is small. A high-V_t assignment to this NMOS transistor will therefore not improve the leakage significantly. Conversely, by setting an NMOS transistor with a very low state probability to high-V_t, the total leakage can be reduced nearly as much as when all transistors in a gate are set to high-V_t, while the impact on performance is much less. Therefore, if we know the state probabilities of the nodes, we can assign high-V_t (or thick T_{ox}) only to those transistors that contribute significantly to the overall leakage, while assigning the rest of the transistors to low-V_t (and thin T_{ox}) to maintain performance. This approach requires that only transistors in a gate with a high probability of being OFF (ON) in are set to high-V_t (thick T_{ox}). Hence, we also propose a new library with carefully selected V_t (and T_{ox}) combinations so that each cell provides favorable trade-offs between leakage and performance under skewed input state probabilities.

We first propose a leakage minimization method for I_{sub} only and then extend it to both I_{sub} and I_{gate}. Starting from the worst performance point (high-V_t or high-V_t / thick-T_{ox} with minimum size transistors) we move to the best performance point by selecting cells from the library with a more speed aggressive V_t (or V_t / T_{ox}) assignment or by increasing the gate size. The sizing and V_t / T_{ox} assignments are performed using a sensitivity-based algorithm where leakage values are calculated using gate input state probabilities. The proposed probability-aware optimization is compared with a traditional optimization which has no information on state probability at nodes (i.e., each node is equally likely to be "0" or "1"). The proposed methods are implemented on synthesized benchmark circuits using an industrial cell library in 0.18um technology for I_{sub} minimization and in a predictive 65nm technology for both I_{sub} and I_{gate} minimization. We show that, in practice, state probabilities are significantly skewed to either a high or low value, making the proposed method effective. On average, the proposed technique improves leakage current by ~30% over a state probability-unaware method with the same cell library options for both I_{sub} and I_{sub} / I_{gate} minimization. The method achieves 62% improvement for I_{sub} minimization and 46% for I_{sub} / I_{gate} minimization if the state probability-unaware method uses a whole gate-based V_t / T_{ox} assignment (i.e., all transistors in a gate are assigned either low-V_t / thin-T_{ox} or high-V_t / thick-T_{ox}). In addition, the total transistor width of the circuit is reduced by approximately 6-10% using the probability-aware method yielding a dynamic power improvement as well.

0-7803-8736-8/05/$20.00 ©2005 IEEE.

2 Leakage Reduction Approach

2.1 Leakage dependence on state probability

In this section we explain how the leakage current can be calculated in runtime mode using state probabilities. It is well known that the leakage current of a gate depends on the input state of that gate. For example, the leakage currents of the simple NAND2 gate have different leakage current values with different input states, as shown in Figure 1. The minimum leakage current at "00" state is only 26% of the maximum leakage current at "10" state.

However, in runtime mode the input state of a gate is unknown. Therefore we compute the leakage current of a gate using the state probabilities of the gate inputs. If we know the state probabilities of the input nodes in a gate we can determine the probability that a gate will be in each state in runtime mode. For example, if two inputs, A and B, for the NAND2 gate in Figure 1 have 0.8 and 0.2 as their state probabilities respectively and assuming that the state probabilities are independent, the probability of the state AB = "10" is P_A x (1 - P_B) = 0.8 x (1 - 0.2) = 0.64. While we assume independent input state probabilities for the purpose of illustration, the implemented computation can account for correlations between the gate input state probabilities using methods in [12].

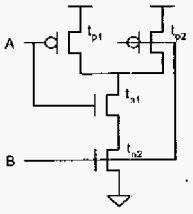
Based on the calculated probability of each state, the leakage current of a gate can be calculated with the below equation:

$$I_{leak} = \sum_{k} P_k \times I_{state,k}$$

In this equation k is over all possible input states in the gate, P_k is the probability of state k and $I_{state,k}$ is the leakage current of state k. If a NAND2 gate in the above example has leakage current values in the table of Figure 1 with given input state probabilities, the leakage current of this NAND2 gate is 189.8 pA.

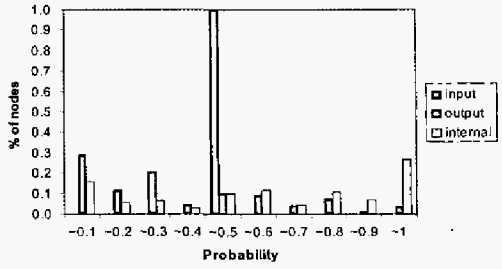
2.2 Input state probability distribution

In this section we demonstrate that node state probabilities show a bi-modal distribution, meaning that some nodes have high state probabilities while other nodes have low state probabilities. This is intuitively clear when we consider the propagation of probabilities through simple logic gates. For instance, if we evaluate a 3-input AND gate where all input have an input state probability of 0.5, the state probability of the gate output is $0.5^3 = 0.125$. Hence, state probabilities tend to diverge to high or low values as they propagate through the circuit. This is also illustrated in Figure 2 which shows the state probabilities of the primary inputs, primary outputs and internal nodes of MCNC benchmark circuit i10. In Figure 2(a) all inputs have P=0.5. However, the state probabilities of outputs and internal nodes are not centered at 0.5 but show a bi-modal distribution. In Figure 2(b), where all inputs have a state probability of either 0.2 or 0.8, the state probabilities of outputs and internal nodes remain lower or higher than those of the inputs. Since the outputs of a circuit will act as inputs to another circuit, it is clear that for such a circuit block the typical state probability for the inputs can be expected to lie in the ranges of P=0.1-0.2 or 0.8-0.9. In our analysis, we therefore use three state probabilities for prime inputs; (1) all inputs have P=0.5, (2) half the inputs have a lower probability of 0.2 and the rest have a higher probability of 0.8 and (3) is identical to (2) except with probabilities P=0.9 and P=0.1.

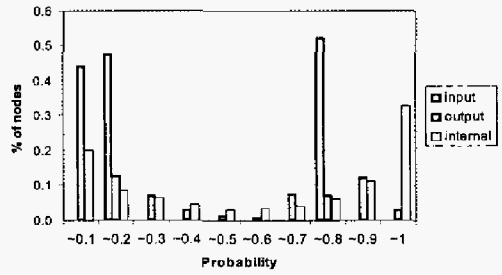


Input state (AB)	Leakage current [pA]
00	71.3
01	267.6
10	235.4
11	106.3

Figure 1. A simple NAND2 gate and its leakage current.



(a) Probabilities of prime inputs = 0.5



(b) Probabilities of prime inputs = 0.2 or 0.8

Figure 2. The state probabilities of i10 circuit

2.3 Probability-aware V_t assignment algorithm

In this section we review how V_t assignment is performed for I_{sub} minimization with a known input state and then show how V_t (or V_t/ T_{ox}) assignment can be combined with the state probabilities for runtime leakage reduction.

We consider the leakage and performance of the simple NAND2 circuit shown in Figure 3. With a known input state, high-V_t assignment is considered only for those OFF transistors that are responsible for I_{sub}. For instance, in state AB = "10" only transistor t_{n2} is considered for high-V_t assignment because assigning other transistors to high-V_t will only decrease the performance of the gate with no reduction in leakage current. Similar to [10], we introduce so-called *groups*, which are the minimum sets of transistors that need to be set to high-V_t to reduce leakage in a particular state. Table 1 shows the leakage current for the NAND2 in Figure 3 for different input states. Column 3 shows the leakage current when high-V_t is assigned to a single group, which is shown in Column 2. Column 4 shows the leak-

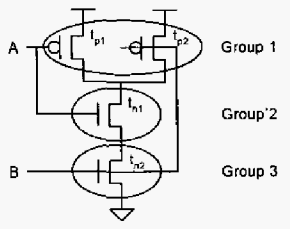


Figure 3. The concept of group at NAND2 gate

Table 1. Leakage current of NAND2 gate

Input State (AB)	Assigned Group	Leakage current [pA]		
		with Group Assign.	with All high-V_t	with All low-V_t
00	3	5.7	4.5	71.3
01	2	16.7	16.7	267.6
10	3	16.5	15.8	235.4
11	1	11.9	11.9	106.3

Table 2. Leakage current for different states and V_t assignments

Input State (AB)	NAND2 Leakage current [pA]				
	All low-V_t	High-V_t at			All high-V_t
		group 1	group 2	group 3	
00	71.3	71.3	14.3	5.7	4.5
01	267.6	267.6	16.7	267.6	16.7
10	235.4	235.4	226.9	16.5	15.8
11	106.3	11.9	106.3	106.3	11.9

age current with all transistors assigned to high-V_t. Leakage current values in these two columns demonstrate that setting only a single group to high-V_t results in equal or nearly equal leakage compared with the leakage when all transistors are assigned high-V_t. At the same time, by setting only a single group to high-V_t, the performance degradation is restricted to only a single signal transition direction and is also reduced compared to high-V_t assignments where most or all transistors are set to high-V_t. Therefore, the leakage reduction and performance penalty trade-off of V_t assignment with a known input state is much improved compared with that with unknown input state.

In runtime mode we can also improve the leakage / performance trade-off with knowledge of state probabilities. Instead of a fixed input state, we exploit the state probability of a gate and combine it with V_t assignment. We first determine the probability of a gate being in a particular state, using the gate input state probabilities, as explained in Section 2.1. In the example of Section 2.1, with $P_A = 0.8$ and $P_B = 0.2$, this NAND2 gate will have a "10" state with probability 0.64. Therefore, the V_t assignment of transistor t_{n2} influences I_{sub} more than any other transistor. This means that if in probability-aware optimization we assign high-V_t to t_{n2}, we can reduce I_{sub} more effectively than when we assign high-V_t to other transistors. Since the remaining transistors are kept at low-V_t, the leakage / performance trade-off is improved compared to when all transistors are assigned high-V_t. On the other hand, if V_t assignment is performed without knowledge of the state probability, each of the states of the gate appears to have equal probability and it is likely that a different group, or possibly the whole gate, is chosen for high-V_t assignment, resulting in a worse actual leakage / delay trade-off. In the above example, if high-V_t is assigned to t_{n2} (group 3) with the given input state probabilities and using the date shown in Table 2, the leakage current of this NAND2 gate is 39.2 pA. If, however for instance group t_{n1} (group 2) is selected for high-V_t assignment due to a lack of state probability information, the leakage becomes 165.2 pA. The leakage current difference between V_t assignment with and without knowledge of the input state probability is therefore ~13X in this example. As shown in Table 3, with given input state probabilities the leakage current with high-V_t at group 3 is the minimum leakage high-V_t assignment whose leakage current (39.2pA) is relatively close to that with all high-V_t (13.4pA). However, cases with high-V_t for group 1 (174.7pA) or for group 2 (165.2pA) show leakage that is closer to that with all low-V_t (189.8pA). This means that without consideration of the input state probabilities, high-V_t assignment to a single group will not improve the leakage / performance trade-off significantly. Hence, it has been common in traditional probability-unaware optimizations to simply assign the entire gate to high or low V_t. On the other hand, with state probability information a group based V_t assignment can significantly improve the leakage current

Table 3. Leakage current with different V_t assignment for NAND2 gate with P_A=0.8, P_B=0.2

NAND2 leakage current [pA]				
All low-V_t	High-V_t at			All high-V_t
	group 1	group 2	group 3	
189.8	174.7	165.2	39.2	13.4

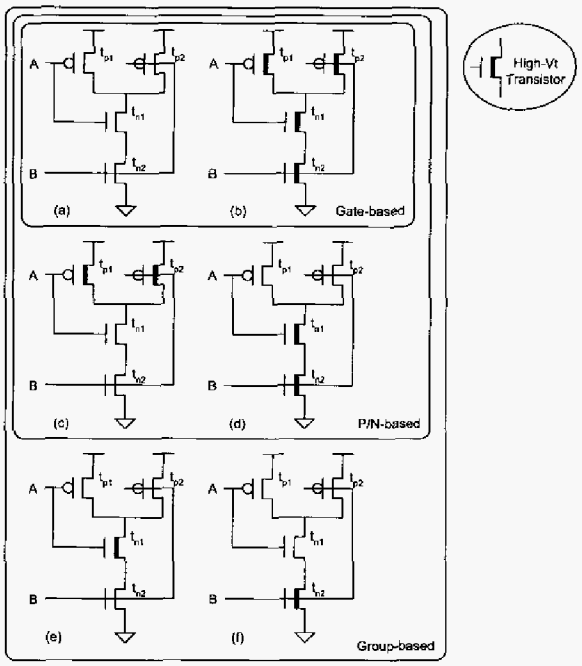


Figure 4. Complete V_t versions of NAND2 gate

and reduce the performance penalty in runtime mode. The same approach can be applied to dual-V_t / dual-T_{ox} assignment for both I_{sub} and I_{gate} minimization. In this case, however, thick-T_{ox} is assigned to transistors that are conducting and exhibit significant I_{gate} current.

2.4 Cell Library Construction

In this section we discuss the construction of needed library cells for dual-V_t or dual-V_t / dual-T_{ox} assignment with consideration of the state probabilities for runtime leakage reduction. In order to perform the leakage current minimization approach by V_t (or V_t / T_{ox}) assignment with knowledge of the state probability, it is necessary to construct a library in which all needed V_t (or V_t / T_{ox}) versions for each cell are available. After such a library has been constructed the process of assigning V_t (or V_t / T_{ox}) can be performed by simply swapping cells from the library.

For V_t assignment only we consider each input state and find the group which is responsible for I_{sub}. As shown in Figure 3 and Table 1, a NAND2 gate has three different groups. In addition to these three groups, the NAND2 gates require two additional cells: all low-V_t transistors for the fastest-performance / highest-leakage and all high-V_t transistors for the slowest-performance / lowest-leakage.

In addition to the above group-based library options, we consider two other library options. The second library option is P/N-based library option, which has only two groups are considered: high-V_t assignment to all PMOS or to all NMOS transistors. This library option is useful in processes where individual V_t assignment is not available in a stack of transistors and also in cases where the input state probabilities are similar among all inputs. The third library option is gate-based, where no group V_t assignment is allowed and gates must consist of either all high or all low V_t transistors. These last cell library versions are useful in two instances: one, when gate input state probabilities are not highly skewed and two, for probability-unaware optimization. The three different library options for NAND2 are shown in Figure 4. Note that the gate-based library is a subset of the P/N-based option, and P/N-based is a subset of group-based.

For simultaneous I_{sub} and I_{gate} minimization a number of different V_t and T_{ox} assignments are possible that provide different leakage / performance trade-off points for different input states. For the fastest

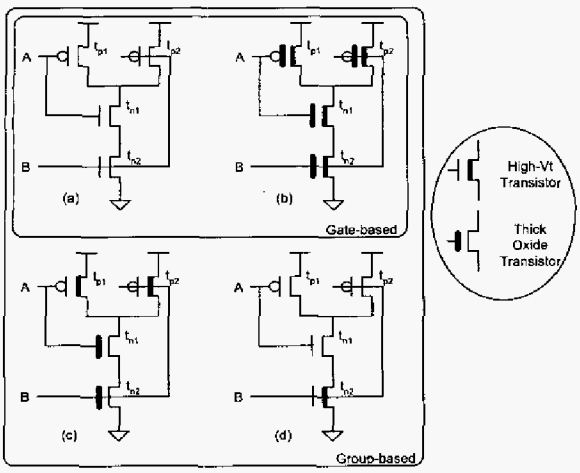


Figure 5. 4-cell V_t-T_{ox} versions of NAND2 gate

delay and highest leakage trade-off point, all transistors are assigned to low-V_t and thin-T_{ox}, such as the NAND2 gate shown in Figure 5(a). On the other hand, for the slowest delay and lowest leakage point over all possible input states, all transistors are assigned to high-V_t and thick-T_{ox} as shown in Figure 5(b). These two cells constitute the gate-based library option, shown in Figure 5 for a NAND2 gate.

In addition to the fastest version and minimum leakage version of the cell, a number of other intermediate trade-off points can be constructed for a cell by assigning only some of the transistors (groups) that contribute to leakage to high-V_t or thick-T_{ox}. For instance, if the NAND2 gate shown in Figure 5(c) has a "11" input state then all transistors affect the leakage current and both NMOS transistors are assigned thick T_{ox} while both PMOS transistors are assigned high-V_t. Therefore, in the "11" state this cell has nearly equal leakage current compared with the case when all transistors are assigned both high-V_t and thick-T_{ox} while having an improved performance. Based on the different library options discussed in more detail in [13], we construct 4-cell versions for each gate. The different V_t-T_{ox} versions of NAND2 gate are shown in Figure 5. Table 4 shows the number of cell versions required for several common gates for both V_t only and V_t / T_{ox} assignments in our runtime leakage minimization.

3 Optimization Approach

In this section we describe the leakage optimization method by V_t (or V_t / T_{ox}) assignment. In addition to the V_t (or V_t / T_{ox}) assignment, we include circuit sizing in order to obtain a better leakage / performance trade-off. The objective of the optimization approach is to achieve the minimum leakage current at a specific delay criterion. Starting from the slowest delay and lowest leakage point with all high-V_t (and thick-T_{ox}) transistors at minimum size, the optimization improves the circuit delay by performing low-V_t (or thin-T_{ox}) assignment and circuit sizing. For V_t (or V_t / T_{ox}) assignment and circuit sizing we use the sensitivity-based method similar to that in [8]. At each iteration during the optimization, the move with the maximum sensitivity value (where a move is either an up-sizing or a low-V_t (or thin-T_{ox}) assignment) is taken to progress from a slow-delay / low-leakage point to fast-delay / high-leakage point. The leakage current values used in calculating the sensitivity values are based on knowledge of the state probability as described earlier.

Table 4. The number of library cells

	V_t only			V_t-T_{ox}	
	Gate-based	P/N-based	Group-based	Gate-based	Group-based
Inverter	2	4	4	2	4
NAND2	2	4	6	2	4
NAND3	2	4	7	2	4
NOR2	2	4	6	2	4
NOR3	2	4	7	2	4

In the V_t (or V_t / T_{ox}) assignment with group-based or P/N-based library options, multiple moves are possible for each gate, while in the gate-based library, only one move needs to be considered for each gate. With a gate-based library, the assignment process moves from the slowest delay / lowest leakage point - all transistors with high-V_t (and thick-T_{ox}) - to the fastest delay / highest leakage point - all transistors with low-V_t (and thin-T_{ox} transistors). However, with a group-based or P/N-based library, an intermediate design point is inserted between these two extreme points. From the all high-V_t (and thick-T_{ox}) version of a gate, high-V_t (or thick-T_{ox}) is first assigned to one of the groups in a gate. After one group is selected for high-V_t (or thick-T_{ox}) assignment, the next move considers setting all transistors to low-V_t (and thin-T_{ox}). In the example of a NAND2 gate with gate-based and group-based libraries shown in Figure 5, the optimization with gate-based library option moves only from (b) to (a). However, if the optimization uses the group-based library, the first step of the optimization is a move from (b) to either (c) or (d), and the second move would be from (c) or (d) to (a).

4 Leakage Model and Characteristics

Since our proposed leakage minimization approach is a library-based method, precharacterized leakage current tables for each library cell are used. Each table has specific leakage current values for each possible input state of a library cell. Based on these current values for each input state, the leakage current value in runtime mode is calculated. For I_{sub} minimization, BSIM3 models from 0.18μm technology were used in SPICE simulation to generate the precharacterized tables. The precharacterized tables for I_{sub} and I_{gate} minimization were constructed using SPICE simulation with BSIM4 models using a predictive 65nm process, which had a gate leakage component that is approximately 36% of the total leakage at room temperature (at which all analyses are performed). For performance characterization, precharacterized delay and output slope tables were stored as a function of cell input slope and output loading. For the 0.18μm technology, low-V_t and high-V_t NMOS (PMOS) devices differ by 14X (10X) in I_{sub} and by 16% (15%) in delay. The difference in I_{gate} for the thick-T_{ox} NMOS devices vs. the thin-T_{ox} device in the 65nm process is 11X whereas I_{sub} is reduced by 17.8X (16.7X) when replacing a low-V_t NMOS (PMOS) device with a high-V_t version in this process. The delay difference between low-V_t / thin-T_{ox} vs. high-V_t / thick-T_{ox} NMOS devices is 70% in 65nm technology.

5 Results

The proposed probability-aware leakage minimization method using V_t (or V_t / T_{ox}) assignment and simultaneous circuit sizing was implemented on a number of benchmark circuits [14] synthesized using an industrial cell library. Based on the given state probabilities of the primary inputs, we compute the state probability of each node in the circuit using the method described in [12]. Our proposed *state probability-aware* method is compared with the *state probability-unaware* method where all nodes have equal probability of 0.5. Our proposed approach is tested with 0.18μm technology for I_{sub} only minimization and 65nm technology for both I_{sub} and I_{gate} minimization, as discussed in Section 4.

A comparison between state probability-unaware and -aware methods for I_{sub} only minimization is shown in Table 5. This optimization performs V_t assignment and sizing with a group-based library. The state probabilities of the primary inputs are P = 0.1/0.9. At three different delay backoff points (5%, 10% and 15% larger delay than minimum achievable delay) the leakage current values with state probability-unaware and -aware methods are shown. The leakage

Table 5. Leakage current comparison between probability-unaware and -aware methods for I_{sub} minimization approach. P=0.1/0.9

	# gates	5% delay penalty			10% delay penalty			15% delay penalty		
		I_{leak} [nA]		Diff. %	I_{leak} [nA]		Diff. %	I_{leak} [nA]		Diff. %
		Un-aware	Aware		Un-aware	Aware		Un-aware	Aware	
i1	41	2.64	0.88	66.5	1.00	0.50	50.1	0.55	0.39	29.1
i2	189	5.28	3.81	27.9	3.51	2.78	20.8	2.55	2.48	3.1
i3	120	5.04	5.01	0.6	3.87	3.09	20.1	3.05	2.51	17.9
i4	160	16.6	9.56	42.2	8.38	6.01	28.3	5.46	4.01	26.5
i5	198	6.12	4.93	19.5	4.67	3.51	24.9	3.85	3.17	17.5
i6	451	16.7	14.4	13.8	8.92	8.65	3.0	7.08	6.77	4.5
i7	547	21.6	12.5	42.2	11.2	8.42	24.9	8.16	7.94	2.7
i8	794	31.2	17.3	44.6	22.6	12.2	46.1	15.5	10.6	31.6
i9	510	31.0	13.0	58.2	15.9	10.1	36.9	10.7	8.35	21.7
i10	1936	34.3	27.5	20.0	26.7	24.6	8.0	24.6	23.2	5.8
alu64	1945	77.3	62.0	19.9	57.0	42.7	25.0	40.9	30.8	24.8
AVG				32.3			26.2			16.8

reduction percentages of the probability-aware method vs. probability-unaware method is also shown. At a relatively tight 5% delay penalty, the leakage current using probability-aware optimization is 32% lower on average with a maximum improvement of 67%. Even at a looser 15% delay penalty point, the probability-aware method has an average 17% lower leakage than the traditional probability-unaware method. Note that for looser delay constraints the difference between the two methods reduces as both methods approach an all high-V_t solution.

In Table 6, we compare the leakage current reduction between different cell library options for I_{sub} only minimization. The leakage current values of the state probability-aware method with gate-based, P/N-based and group-based libraries are compared with that of state probability-unaware method with a gate-based library option where the performance-leakage trade-off is the worst among the different library options. Table 6 shows that with the same gate-based library, the probability-aware method has 26% lower leakage current on average than the probability-unaware method (Column 4). When the probability-aware method utilized a P/N-based library and the group-based library, this leakage improvement increases to 55% and 62% leakage reduction, respectively. This shows that the probability-aware method benefits significantly from these library options that were specifically tailored for skewed input probabilities.

Table 7 provides a comparison of results between state probability-unaware and -aware methods for both I_{sub} and I_{gate} minimization. The state probabilities of the primary inputs are P = 0.1/0.9. Note that since the maximum delay difference for this dual-V_t / T_{ox} 65nm technology is much larger than that in the dual-V_t 0.18μm technology, as discussed in Section 4, we use different delay backoff points. Across the three delay penalty points, the probability-aware method shows approximately 30% lower leakage current on average than the probability-unaware method, with a 73% maximum improvement.

Table 7. Comparison between probability-unaware and -aware minimization for both I_{sub} and I_{gate}. Group-based library, P=0.1/0.9

	20% delay penalty			30% delay penalty			40% delay penalty		
	I_{leak} [μA]		Diff. %	I_{leak} [μA]		Diff. %	I_{leak} [μA]		Diff. %
	Unaware	Aware		Unaware	Aware		Unaware	Aware	
i1	4.26	2.61	38.8	3.62	1.48	59.0	1.97	0.8	59.6
i2	19.3	16.8	13.2	13.2	12.9	2.2	11.6	9.26	20.5
i3	24.4	22.9	6.2	19.0	17.8	6.2	15.7	12.3	21.2
i4	25.6	20.5	19.9	20.0	12.6	37.0	11.3	9.9	12.3
i5	14.6	12.1	17.0	10.1	9.0	10.7	7.61	6.37	16.3
i6	78.2	66.8	14.6	58.6	41.5	29.2	50.2	19.7	60.7
i7	132.3	98.2	25.8	119.2	44.7	62.5	94.7	25.5	73.0
i8	106.6	39.4	63.0	79.1	26.6	66.4	47.7	22.8	52.3
i9	92.1	39.5	57.1	73.7	35.4	52.0	35.1	23.3	33.6
i10	92.1	64.5	30.0	61.7	41.2	33.2	46.9	32.5	30.8
alu64	129.7	108.8	16.1	88.5	76.3	13.8	65.5	56.6	13.6
AVG			27.4			33.8			35.8

A comparison of the leakage current reduction between different cell library options for both I_{sub} and I_{gate} minimization is shown in Table 8. The leakage current values of state probability-aware optimization with gate-based and group-based libraries are compared with that of the state probability-unaware method with a gate-based library option which resulted in 20% and 47% less leakage than the probability-unaware minimization, respectively.

Table 9 shows the comparison between different state probabilities of the primary inputs. When the primary inputs show moderate state probabilities of 0.5, the probability-aware optimization performs at its worst but still enables 15% leakage reduction compared to probability-unaware techniques. This indicates that the proposed techniques are universally applicable and can be useful even when node state probabilities are not divergent as described earlier.

In Table 10, we compare the total transistor size (width) of the circuits between state probability-unaware and -aware methods for I_{sub} only minimization using a group-based library with state probabilities of the primary inputs P=0.1/0.9. The results show that our proposed method results in a smaller circuit size than the probability-unaware method by 6-10% on average. Since dynamic power is proportional to total transistor width, the proposed state probability-aware leakage optimization method results in lower dynamic power

Table 6. Leakage current comparison between cell library options for I_{sub} minimization approach. 5% delay penalty point, P=0.1/0.9

	Probability-unaware	Probability-aware method (current in nA) (Diff% vs. unaware & gate-based)					
	Gate-based library	Gate-based library		P/N-based library		Group-based library	
	I_{leak}	I_{leak}	Diff. %	I_{leak}	Diff. %	I_{leak}	Diff. %
i1	1.01	0.94	6.8	1.03	-1.6	0.88	12.5
i2	7.7	6.4	16.9	4.65	39.7	3.81	50.6
i3	9.59	9.05	5.7	5.24	45.32	5.01	47.74
i4	33.64	31.6	6.2	14.6	56.5	9.56	71.6
i5	12.8	12.3	4.1	5.98	53.2	4.93	61.5
i6	60.0	14.5	75.8	15.1	74.9	14.4	76.0
i7	108.0	42.5	60.7	20.1	81.4	12.5	88.4
i8	82.8	47.7	42.4	17.1	79.3	17.3	79.1
i9	119.8	55.5	53.6	18.8	84.3	13.0	89.2
i10	47.3	43.4	8.2	29.0	38.6	27.5	41.9
alu64	187.6	175.1	6.7	78.8	58.0	62.0	67.0
AVG			26.1		55.4		62.3

Table 8. Comparison between cell library options for I_{sub} and I_{gate} minimization. 20% delay penalty point, P=0.1/0.9 (current in μA)

	Probability-unaware	Probability-aware method (Diff% vs. unaware & gate-based)			
	Gate-based library	Gate-based library		Group-based library	
	I_{leak}	I_{leak}	Diff. %	I_{leak}	Diff. %
i1	4.6	4.6	0.0	2.61	43.2
i2	29.1	28.4	2.5	16.8	42.4
i3	38.4	31.5	18.0	22.9	40.5
i4	39.5	37.6	4.5	20.5	48.2
i5	21.1	17.0	19.4	12.1	42.6
i6	90.8	74.8	17.6	66.8	26.5
i7	140.5	85.7	39.0	98.2	30.1
i8	145.1	75.0	48.3	39.4	72.8
i9	177.9	84.7	52.4	39.5	77.8
i10	122.2	122.1	0.04	64.5	47.2
alu64	185.2	162.8	12.1	108.8	41.3
AVG			19.5		46.6

Table 9. Leakage current comparison between state probabilities of the prime inputs using group-based libraries.

	Leakage current difference % between unaware and aware					
	I_{sub} minimization (at 5% delay penalty point)			I_{sub} & I_{gate} minimization (at 20% delay penalty point)		
	P=0.1/0.9	P=0.2/0.8	P=0.5	P=0.1/0.9	P=0.2/0.8	P=0.5
i1	66.5	68.0	45.4	38.8	33.2	25.7
i2	27.9	10.8	19.4	13.2	21.0	8.6
i3	0.6	3.0	0.0	6.2	1.1	0.5
i4	42.2	23.6	5.3	19.9	17.3	11.9
i5	19.5	9.4	0.4	17.0	9.4	0.3
i6	13.8	10.3	0.4	14.6	20.4	30.7
i7	42.2	28.0	10.7	25.8	23.8	23.4
i8	44.6	38.1	33.5	63.0	53.4	42.4
i9	58.2	49.7	35.4	57.1	39.4	26.2
i10	20.0	14.4	7.6	30.0	29.0	18.5
alu64	19.9	7.3	2.4	16.1	7.7	4.7
AVG	32.3	23.9	14.6	27.4	23.3	17.5

Table 10. Total transistor size comparison between state probability-unaware and -aware methods for I_{sub} minimization approach. (group-based library, P=0.1/0.9)

	5% delay penalty			10% delay penalty			15% delay penalty		
	Width[mm]		Diff. %	Width[mm]		Diff. %	Width[mm]		Diff. %
	Unaware	Aware		Unaware	Aware		Unaware	Aware	
i1	0.36	0.35	1.6	0.35	0.3	15.2	0.3	0.29	5.2
i2	2.3	2.02	11.9	2.0	1.81	9.9	1.8	1.72	4.2
i3	1.57	1.38	12.4	1.46	1.16	20.6	1.34	1.1	18.2
i4	1.51	1.5	0.6	1.45	1.38	4.6	1.37	1.28	6.6
i5	1.08	0.99	8.4	0.94	0.89	5.3	0.9	0.8	11.0
i6	3.39	3.27	3.7	2.9	2.79	3.82	2.62	2.46	6.0
i7	4.1	3.95	3.7	3.59	3.14	12.6	3.18	2.95	7.1
i8	6.18	6.06	2.0	5.6	4.74	15.4	5.01	4.17	16.8
i9	6.25	5.01	19.8	5.45	4.31	20.9	4.46	3.57	19.9
i10	10.9	10.1	7.4	9.96	9.5	4.6	9.47	8.97	5.3
alu64	11.5	11.2	2.7	10.7	11.1	-3.2	10.0	10.4	-4.1
AVG			6.7			10.0			8.7

as well as reduced static power over traditional techniques.

Finally, Figure 6 plots the leakage current results for the proposed method as well as the probability-unaware method, with different library options as a function of the delay for circuit i10. As shown in Figure 6, the proposed probability-aware method with group-based library has the lowest leakage current. Since the gate-based libraries consistently show the highest leakage current for a given delay, we can say that the use of group-based (or at least P/N based) libraries is critical to leakage current optimization, as is the use of state probabilities. In Figure 6, we also see that the group-based library option shows the biggest difference between the probability-aware and -unaware methods.

6 Conclusions

In this paper, we have proposed a new leakage optimization method that specifically targets runtime leakage current. The method utilizes the skewed gate input state probabilities by setting only those transistors in a gate to high-V_t that are most likely to be contribute significantly to the total leakage current. The approach uses a sensitivity based approach where leakage current is computed using the gate input state probabilities. A library where V_t and T_{ox} assignment with favorably trade-offs under skewed input probabilities are available was developed and results in significant improvement in the leakage reduction of the probability-aware optimization approach. Compared with a probability-unaware optimization that is gate-based V_t, the proposed method improves run time leakage current by 62%, on average, over all benchmark circuits and with a maximum improvement of 89%.

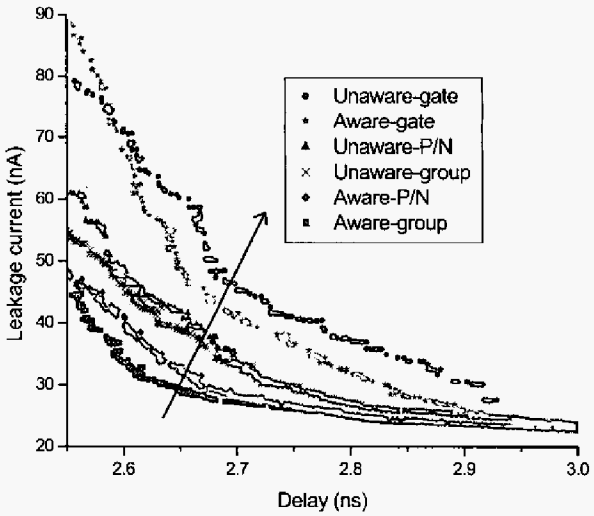


Figure 6. Leakage current comparison for i10

References

[1] S. Narendra, *et al.*, "Leakage issues in IC design: trends, estimation and avoidance", Proc. ICCAD (tutorial), 2003.

[2] S. Mutoh, *et al.*, "1-V power supply high-speed digital circuit technology with multithreshold-voltage CMOS," IEEE JSSC, Aug. 1995.

[3] S. Shigematsu, *et al.*, "A 1-V high-speed MTCMOS circuit scheme for power-down application circuits," IEEE Journal of Solid-State Circuits, vol. 32, pp. 861-869, June 1997.

[4] J. Halter and F. Najm, "A gate-level leakage power reduction method for ultra-low-power CMOS circuits," Proc. CICC, pp. 475-478, 1997.

[5] V. De, *et al.*, "Techniques for leakage power reduction," in *Design of high-performance microprocessor circuits*, IEEE press, 2001.

[6] M.C. Johnson, *et al.*, "Models and algorithms for bounds on leakage in CMOS circuits," IEEE Trans. CAD, pp. 714-725, June 1999.

[7] L. Wei, *et al.*, "Design and optimization of low voltage high performance dual threshold CMOS circuits," Proc. DAC, 1998.

[8] S. Sirichotiyakul, *et al.*, "Duet: an accurate leakage estimation and optimization tool for dual Vt circuits," IEEE Trans. VLSI, April 2002.

[9] M. Ketkar and S. Sapatnekar, "Standby power optimization via transistor sizing and dual threshold voltage assignment," Proc. ICCAD, 2002.

[10] D. Lee and D. Blaauw, "Static leakage reduction through simultaneous threshold voltage and state assignment," Proc. DAC, pp.191-194, 2003.

[11] G. Sery, *et al.*, "Life is CMOS: Why chase the life after?," Proc. DAC, pp.78-83, 2002.

[12] S. Ercolani, *et al.*, "Estimate of signal probability in combinational logic networks," Proc. European Test Conference, 1989, pp. 132 - 138.

[13] D. Lee, *et al.*, "Simultaneous state, Vt and Tox assignment for total standby power minimization," Proc. Design, Automation and Test in Europe Conference and Exhibition, pp.494-499, 2004.

[14] http://www.cbl.ncsu.edu