Minimizing Leakage of Sequential Circuits through Flip-Flop Skewing and Technology Mapping

Sewan Heo and Youngsoo Shin

***Abstract*—Leakage current of CMOS circuits has become a major factor in VLSI design these days. Although many circuit-level techniques have been developed, most of them require significant amount of designers' effort and are not aligned well with traditional VLSI design process. In this paper, we focus on technology mapping, which is one of the steps of logic synthesis when gates are selected from a particular library to implement a circuit. We take a radical approach to push the limit of technology mapping in its capability of suppressing leakage current: we use a probabilistic leakage (together with delay) as a cost function that drives the mapping; we consider pin reordering as one of options in the mapping; we increase the library size by employing gates with larger gate length; we employ a new flip-flop that is specifically designed for low-leakage through selective increase of gate length. When all techniques are applied to several benchmark circuits, leakage saving of 46% on average is achieved with 45-nm predictive model, compared to the conventional technology mapping.**

***Index Terms*— Low power, leakage current, logic synthesis, technology mapping, VLSI design**

I. INTRODUCTION

Scaling down of transistors has resulted in dramatic increase of leakage current. Threshold voltage of MOSFET devices has been scaled down to compensate for the reduced circuit performance in low supply voltage, which causes exponential increase of subthreshold leakage. Gate oxide has been scaled down as well for better control of MOSFET channel current, which leads to large amount of gate leakage. The leakage current, in fact, has become a major portion of total power consumption, and, in many technologies, it contributes up to 50% of the overall power consumption [1].

Many circuit-level techniques have been proposed to control leakage such as power gating, body bias, input vector control, selective MTCMOS, zigzag power gating, mixed V_t, and so on [1]. However, most of these techniques require significant amount of designers' effort during design process and the support of dedicated design tools. These are some of reasons why these techniques are not yet prevalent in large scale circuit design.

In this paper, we focus on technology mapping, which is one of the steps of logic synthesis when gates are selected from a particular library to implement a circuit. The technology mapping takes an optimized logic network (as a result of technology independent logic minimization) as its input and outputs a netlist of gates, which minimizes a total cost (usually area, delay, or the combination of the two). Since the technology mapping is the only step in logic synthesis where the detailed leakage information is available, we try to take a radical approach to see how much leakage we can save while timing constraints are still satisfied. We use a weighted sum of probabilistic leakage and delay as a cost function of the mapping as opposed to traditional area and delay metrics. We consider pin reordering as one of the options in the mapping. We increase the library size by

Manuscript received Nov. 5, 2007; revised Nov. 30, 2007

Department of Electrical Engineering
KAIST, Daejeon 305-701, Korea
E-mail : hsewan@kaist.ac.kr, youngsoo@ee.kaist.ac.kr

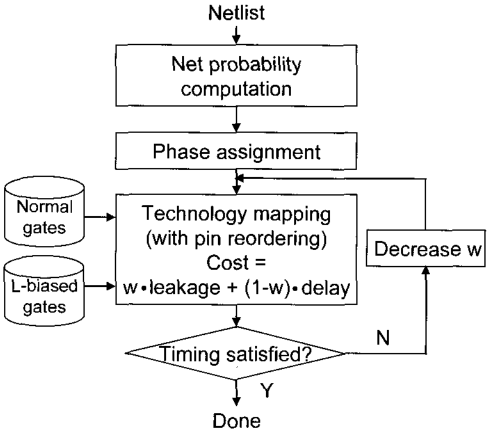

Netlist
Net probability computation
Phase assignment
Normal gates
L-biased gates
Technology mapping (with pin reordering) Cost = w•leakage + (1-w)•delay
Decrease w
Timing satisfied?
N
Y
Done

Fig. 1. Overall flow of the proposed technology mapping.

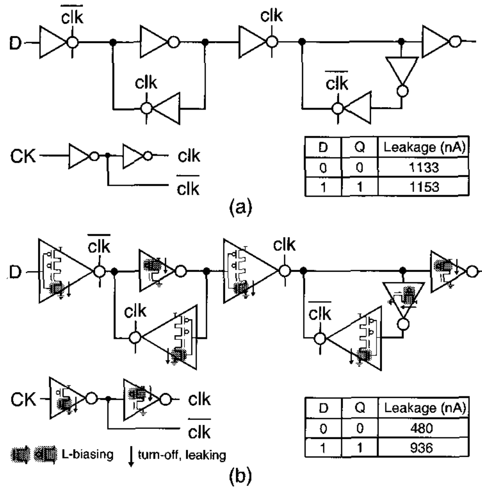
employing gates with larger gate length, thus less leakage with slight increase of delay. We employ a new set of flip-flops that are specifically designed for low-leakage through selective increase of gate length. Depending on the state probability of each flip-flop, we either choose the gate-length-biased flip-flop or the one with its state complemented. The prototype tool was implemented in SIS [6] logic synthesis environment. The results with several benchmark circuits show that we can reduce leakage by 46% on average in 45-nm predictive technology model.

The remainder of this paper is organized as follows. In the next section, we briefly explain gate-length biasing and pin reordering, which are two main techniques we use in the technology mapping, followed by the overall flow of our mapping procedure. In Section III, we propose a gate-length-biased flip-flop, which has characteristics of unequal leakage and delay, and phase assignment procedure that exploits these flip-flops. Experimental results with several benchmark circuits are presented in Section IV, and we draw conclusion in Section V.

II. PRELIMINARIES

2.1 Gate-Length Biasing

Gate-length biasing involves a small increase in the gate lengths of devices. In a 130-nm industrial process, it is reported [2] that an 8 nm increase in gate length yields 30% decrease in leakage with 5% increase in delay for a minimum size inverter. This large decrease in leakage with just a small increase of delay occurs because the nominal gate length of the technology is usually very close to the knee of the leakage versus gate length curve that is produced by short channel effects. This small increase in gate length does not affect the printability during the manufacturing process, and can usually allows pin compatibility with the unbiased version of the cell, which benefits post placement optimization.



D	Q	Leakage (nA)
0	0	1133
1	1	1153

D	Q	Leakage (nA)
0	0	480
1	1	936

Fig. 2. An example D flip-flop: (a) original and (b) gate-length biased one.

In addition to a set of gates with nominal gate length, we have the same set of gates with larger gate length as shown in Fig. 1. For sequential elements such as flip-flops, we apply gate-length biasing, but only to a subset of transistors, which will be explained in Section III.

The additional set of gates created by exploiting gate-length biasing enlarges search space during synthesis. They may used for low leakage library, as long as the small delay increase does not induce a critical timing problem.

2.2 Pin Reordering

Pin reordering refers to exchanging the inputs of a gate when they are compatible [3]. Take an example of two-input NAND gate with inputs A and B (with A being closer to the output). If the signal probability of B is higher than that of A, exchanging the two inputs can help reduce gate leakage, since the nMOS device connected to ground can be a main source of gate leakage when its gate terminal (B) is driven by the signal of high probability of being one.

Furthermore, when combined with gate-length biasing, pin reordering can lead to a substantial reduction of gate leakage, since subthreshold leakage can be reduced by proper gate-length biasing. Our experiments reveal that about 80% of leakage can be reduced in four-input NAND gates via combined pin reordering and gate-length biasing.

Using pin reordering causes almost no penalty. Since the exchanged inputs are logically the same with the original one, the technique can be readily implemented in the conventional synthesis environment and does not require additional cost for manufacturing. Furthermore, subthreshold leakage through transistor stack is not affected by proper signal probability reordering. Therefore, pin reordering can be used simultaneously with gate-length biasing for further leakage reduction.

2.3 Overall Flow

Fig. 1 shows the overall flow of the proposed technology mapping. It takes a logic network of a sequential circuit, which represents multiple Boolean functions (i.e. flip-flop input functions and circuit output functions), as its input, and generates a gate-level netlist, where gates are selected from a technology library. In the library, we assume gates with larger gate length in addition to those with nominal gate length.

In order to obtain a state probability (i.e. probability of Q-output of each flip-flop being logic one), we simulate the network with a sequence of sample input patterns, monitor the Q-outputs, and derive their probabilities. These probabilities, together with the signal probabilities of primary inputs, are propagated through the network [4] to obtain signal probabilities of all the nets. These probabilities are then used to derive the leakage of any gate that is to be mapped on the network.

Before we start the mapping of combinational subcircuit, we go through a step, which we call *phase assignment*. In this step, we try to minimize the leakage of flip-flops, which will be explained in detail in the next section.

For technology mapping, each function in the network is represented as a set of base functions[1], which is called a subject graph. Each gate in the library is likewise represented using the base function, which are called pattern graphs. The technology mapping, thus, is to find an optimal-cost covering of subject graphs using the collection of pattern graphs [5]. Since general covering is not likely to be solved in reasonable amount of time, it is approximated as a series of tree covering. The tree covering can be solved in polynomial time via dynamic programming. The cost function we use in the dynamic programming is a weighted sum of leakage and delay (as opposed to conventional area and/or delay) as indicated in Fig. 1. Note that the leakage is computed from the signal probabilities of the nets. For example of two-input NAND gate, its leakage can be expressed by:

$$L = l_{00}(1-P_A)(1-P_B) + l_{01}(1-P_A)P_B$$
$$+ l_{10}P_A(1-P_B) + l_{11}P_A P_B,$$

where P_A and P_B denote the signal probability of input-A and input-B, respectively. The l_{ij} corresponds to the leakage of the gate when input-A is logic i and input-B is logic j.

We consider the possibility of pin reordering when we consider the candidates for the mapping. The weight for the leakage (ω) is initially 1.0 implying that we try to find the mapping that leads to minimum leakage. If the timing is not satisfied, we decrease ω and try another mapping. The procedure is iterated until the timing constraints are satisfied, which guarantees the minimum leakage within timing constraint.

III. PHASE ASSIGNMENT

3.1 Gate-Length Biased Flip-Flop

Fig. 2(a) shows an example D flip-flop with inverter and tristate inverter implementation. Over the operation of flip-flops, both D-input and Q-output have the same logic state most of the time, since a new D-input which is one of the outputs of combinational subcircuit (and arrives shortly before active clock edge) will be captured

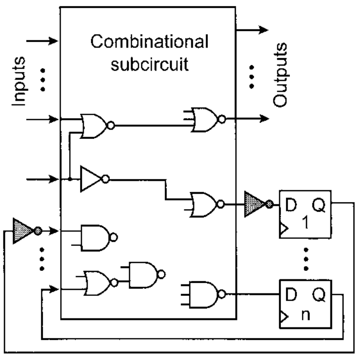


Fig. 3. Assignment of complemented state.

[1] Base functions are a set of gates that can implement all the Boolean functions. An example is an inverter and two-input NAND gate.

and propagated to the Q-output at active clock edge. The leakages for two possible flip-flop states are also shown in Fig. 2(a), which indicate that the leakage is almost independent of the state (for this particular flip-flop).

However, if we employ gate-length biasing for the transistors that are turned off when both D-input and Q-output are logic low as shown in Fig. 2(b), the leakage for the two flip-flop states can be made very different as shown in the figure. Specifically, if we increase the gate length of the transistors, which are marked in Fig. 2(b), the leakage when D-input and Q-output are logic low becomes 480 nA as opposed to the original 1133 nA. The leakage when D-input and Q-output are logic high is also reduced (from 1153 nA to 936 nA), mainly due to the two gate-length-biased transistors in the cascaded inverters, which are responsible for generating internal clock signals.

The benefit of leakage reduction from the gate-length-biased flip-flops is considerable in sequential circuits, since a large portion of total leakage is from sequential elements as shown in section IV.

The gate-length-biased flip-flop has skewed timing parameters. The rising and falling clock-to-Q delay is increased by 32% and 7%, respectively. The increase of rising delay is larger than that of falling delay since the transistors whose gate length is increased are sensitized for rising signal. The rising and falling setup time is increased by 34% and 24%, respectively.

3.2. Phase Assignment of Flip-Flop

Since the leakage of gate-length-biased flip-flops is very different for different flip-flop states, it can be exploited during the technology mapping as shown in Fig.1 (the box named phase assignment). If the state probability is higher than 0.5, we want to have the state complemented, so that it has more chance to remain in low leakage state (both D and Q are logic low). This can be accomplished as follows. As an example of a sequential circuit as shown in Fig. 3, suppose we want to complement the state of the first D flip-flop. We simply insert two inverters: one before the D-input and the other after Q-output. The second inverter can be avoided if $\overline{Q}$ is available, since we can achieve the same goal by swapping Q and $\overline{Q}$. The extra inverters, if left, may not be an overhead, since they can be absorbed in the combinational subcircuit and, after its mapping, they are likely to disappear. The same holds for other types of flip-flops. For example of J-K flip-flop, it can be readily shown that by exchanging J and K inputs and Q and $\overline{Q}$ outputs, respectively , we can complement the original flip-flop state.

Table 1. Total leakage reduction with benchmarks.

Circuit	#gates	#F/Fs	L-D map	+Logic Opt	+F/F Opt.
s349	550	15	9.4%	33.5%	48.1%
s382	735	21	11.2%	20.6%	46.0%
s386	924	6	6.4%	24.4%	31.9%
s400	788	21	15.5%	27.5%	53.1%
s510	1103	6	6.7%	31.8%	37.5%
s641	885	19	16.7%	35.1%	51.9%
s713	953	19	14.2%	33.8%	49.0%
s838	1891	32	11.9%	30.7%	49.4%
s1423	2492	74	4.1%	19.3%	44.5%
s1488	3555	6	26.5%	48.7%	50.8%
s1494	3606	6	18.9%	43.5%	45.9%
Average			12.9%	31.7%	46.2%

For flip-flops with state probability less than 0.5, we simply use gate-length-biased flip-flops (as far as timing of the circuit is satisfied) without complementing their states.

The phase assignment is more efficient when used with gate-length-biased flip-flops and used in control path (with most of signal probabilities far from 0.5) rather than data path. Since it is based on probabilistic flip-flop state, complementing state is more powerful when flip-flops can be in a low-leakage state with high probability.

IV. Experimental Results

We performed experiments on a set of circuits taken from the MCNC and ISCAS'89 benchmarks. Each

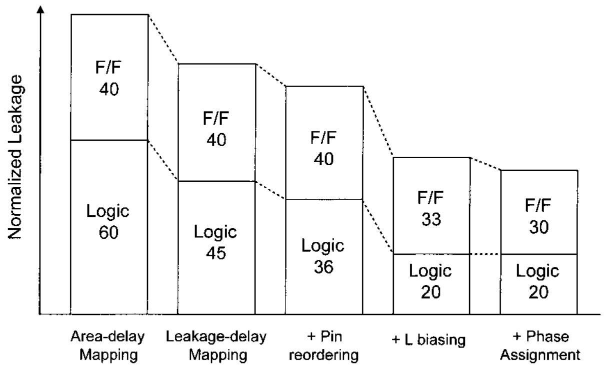


Fig. 4. Leakage reduction of s382 by each technique.

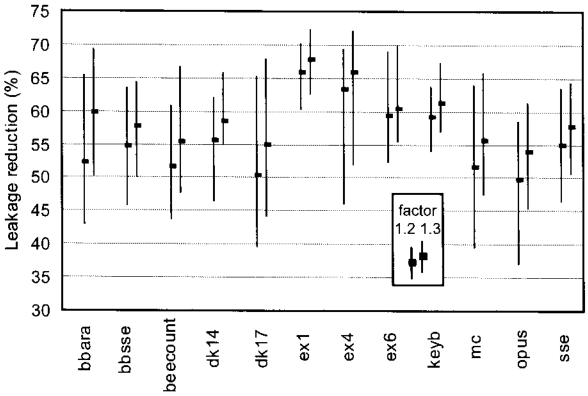


Fig. 5. Variation of leakage saving for varying input probabilities and varying timing constraints.

circuit was synthesized with SIS [6] and mapped into a gate library, which we built for 45-nm predictive model [7]. The proposed technology mapping was implemented in SIS [6] environment as well.

Shown in the first three columns of Table I are the name of the circuits, the number of gates in the combinational subcircuit, and the number of flip-flops. In the fourth column, we see the amount of leakage saving when we use a cost function of weighted sum of leakage and delay (refer to Fig. 1) compared to leakage when conventional cost function of area and delay is used. For each circuit, we assume 1.5 times of critical path delay (when we map the circuit with cost function of delay alone) as its timing constraint. We see about 13% saving on average. When we employ pin reordering and library of gate-length-biased gates to our mapping, the total saving increases to about 32% on average as shown in the fifth column, implying that combined pin reordering and gate-length biasing alone yields about 19% of leakage saving. After we employ phase assignment of flip-flops, the overall saving even goes up to 46% on average (sixth column), which is significant.

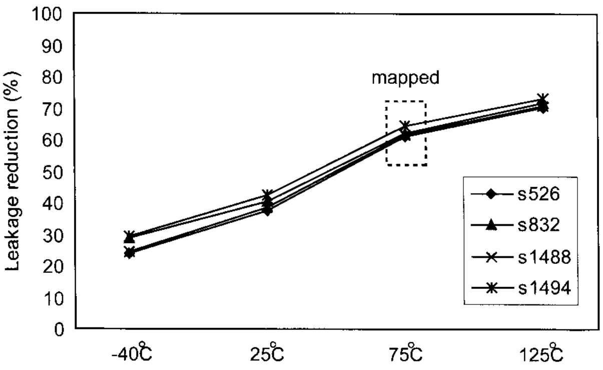


Fig. 6. Variation of total leakage reduction with temperature.

The effect of each technique for leakage reduction is analyzed with an example circuit s382 in Fig. 4. The leakage is normalized to the total leakage of the circuit when it is mapped with conventional cost function of area and delay (leftmost bar). The effect of the mapping with a cost function of leakage and delay is reflected in the second bar. The effect of pin reordering in the combinational subcircuit alone is shown in the third bar. The fourth bar represents the effect of gate-length-biasing on sequential as well as combinational portion of the circuit. The last bar indicates the effect of phase assignment.

The total leakage is reduced by about 50% by all techniques applied simultaneously during technology mapping. Leakage of combinational subcircuit is reduced by pin reordering and gate-length biasing with leakage-delay mapping, while that of flip-flops is reduced by phase assignment with skewed flip-flop. This is from reduction of subthreshold leakage and gate leakage by gate-length biasing and pin reordering, respectively.

Since our technology mapping is driven by input signal probabilities, which can vary over execution of circuits, it is important to guarantee sizable leakage saving even though there is a variation of input signal probabilities. Fig. 5 shows the variation of leakage saving of MCNC benchmark circuits for different input signal probabilities. Each bar represents a range of leakage saving under 100 different average input probabilities of circuit inputs. The dot in each bar indicates the average leakage saving. We also repeat the same experiment for different timing constraints. The timing constraint of each circuit is assumed 1.2 and 1.3 times, respectively, of critical path delay (when we map the circuit with cost function of delay alone). As we allow loose timing constraint, the leakage saving is increased, as it must.

Since our mapping involves leakage, which is a function of temperature, and the mapping is performed for fixed temperature, while temperature itself varies over time, it is important to ensure that the mapping is not too sensitive to temperature. We take four example circuits, map them at fixed temperature, and simulate them to see their leakage saving while we vary temperature, as shown in Fig. 6. The leakage saving increases with temperature, as expected. At higher temperature, the circuits are more leaky (by dominant subthreshold leakage) and gate-length biasing is more effective, which governs the leakage saving. As temperature is decreased, the absolute leakage itself is decreased (by steady gate leakage and exponentially decreased subthreshold leakage), and pin reordering is a main driver for leakage saving.

V. CONCLUSIONS

Although many circuit techniques have been proposed, they do not align well with conventional VLSI design due to many custom engineering. In this paper, we proposed leakage-aware technology mapping, which is one of steps of logic synthesis and is usually transparent to designers. We tried every efforts to push the limit of capability of technology mapping in terms of leakage saving. We used a probabilistic leakage (together with delay) as a cost function that drives the mapping; we considered pin reordering as one of the options in the mapping; we increased the library size by employing gates with larger gate length; we employed a new flip-flop that is specifically designed for leakage through selective increase of gate length. When all techniques are applied during technology mapping, an average leakage saving of 46% was achieved, compared to the conventional technology mapping.

REFERENCES

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

[2] P. Gupta, A. B. Kahng, P. Sharma, and D. Sylvester, "Selective gatelength biasing for cost-effective runtime leakage control," in *Proc. Design Automat. Conf.*, June 2004, pp. 327-330.

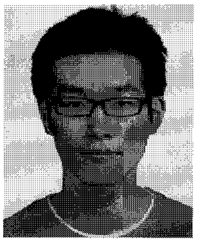
[3] D. Lee, W. Kwong, D. Blaauw, and D. Sylvester, "Analysis and minimization techniques for total leakage considering gate oxide leakage," in *Proc. Design Automat. Conf.*, June 2003, pp. 175-180.

[4] S. Ercolani, M. Favalli, M. Damiani, P. Olivo, and B. Ricc′o, "Estimate of signal probability in combinational logic networks," in *Proc. European Test Conf.*, Apr. 1989, pp. 132-138.

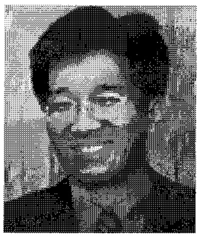
[5] K. Keutzer, "DAGON: technology binding and local optimization by DAG matching," in *Proc. Design Automat. Conf.*, June 1987, pp. 341-347.

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

[7] W. Zhao and Y. Cao, "New generation of predictive technology model for sub-45nm design exploration," in *Proc. Int'l Symp. on Quality Electronic Design*, Mar. 2006, pp. 585-590.



Sewan Heo was born in Busan, Republic of Korea, 1983. He received the B.S. and M.S. degree in the Department of electrical engineering from Korea Advanced Institute of Science and Technology (KAIST), Korea, in 2005 and 2007, respectively. He is currently working as a researcher in Electronics and Telecommunications Research Institute (ETRI), Korea, from 2007. His research interests are in low power digital circuit design and multimedia processor design.



Youngsoo Shin received the B.S., M.S., and Ph.D. degrees in electronics engineering from Seoul National University, Korea. He has worked at the University of Tokyo, Japan, as a Research Associate, and IBM T. J. Watson Research Center, Yorktown Heights, NY, as a Research Staff Member. He is currently an Associate Professor in the Department of Electrical Engineering, KAIST, Daejeon, Korea. He received a Best Paper Award at the 2005 Int'l Symp. on Quality Electronic Design (ISQED). He has been on the program committee for Int'l Symp. on Low Power Electronics and Design (ISLPED), Int'l Conf. on Computer-Aided Design (ICCAD), and Asia and South Pacific Design Automation Conf. (ASP-DAC).