Optimization Technique for FB/TB Assignment in PD-SOI Digital CMOS Circuits

Rafik S. Guindi, Roman C. Kordasiewicz, and Farid N. Najm
University of Toronto, Toronto, ON, M5S-3G4, Canada
rguindi@eecg.utoronto.ca, roman.kordasiewicz@utoronto.ca, f.najm@utoronto.ca

Abstract—**This work presents a technique for reducing the total leakage current in PD-SOI combinational circuits by mixing floating-body and tied-body transistors in the same circuit. Basic gate characterization data are first presented, and then used as part of a static timing analysis based optimization algorithm. Results obtained from a number of benchmark circuits show a decrease of up to 86% in total leakage current.**

I. Introduction

In recent years, silicon-on-insulator (SOI) has emerged as a promising alternative to bulk-silicon CMOS technology for low-power high-performance applications [1], [2]. Partially depleted SOI (PD-SOI) technology provides a degree of freedom not found in traditional bulk-Si CMOS in that devices are manufactured with separate bodies isolated from one another, with the potential for individual body voltages to take on different values. Devices can be manufactured to have either a floating body (FB), or a tied body (TB) with a body contact that can be connected to a particular voltage. The speed advantage offered by PD-SOI technology is mostly associated with the use of FB devices. When a voltage transition occurs at the gate of an FB MOS transistor, the body voltage follows the gate voltage due to capacitive coupling, dynamically changing the value of the threshold voltage V_T, resulting in a faster device. This phenomenon however is responsible for increased subthreshold leakage in FB transistors. When leakage current is a critical concern, TB devices outperform their FB counterparts, since, by rail-tying the transistor body, V_T can be fixed at its largest value, and the subthreshold leakage reduced. On the other hand, TB devices consume significantly more area than FB devices because each transistor requires its own body contact.

In this work, we take advantage of the tradeoff between speed, subthreshold leakage, and area, by mixing FB and TB devices in the same CMOS combinational circuit. We aim at reducing the total leakage current in a large circuit by strategically using TB devices along signal paths that are off the circuit's critical path, without compromising the overall speed advantage offered by the PD-SOI technology.

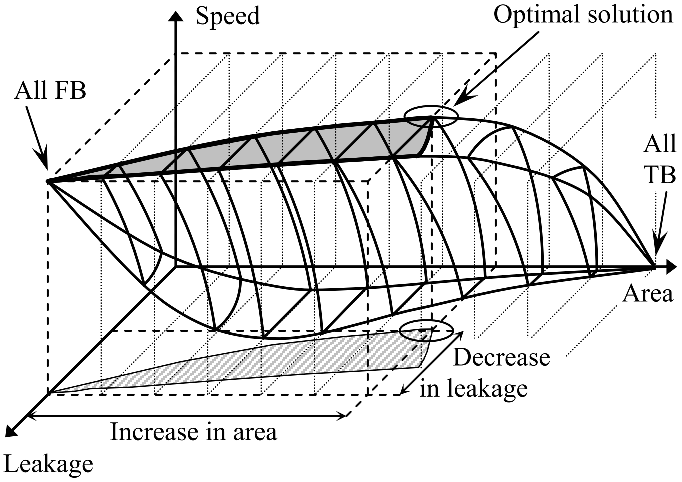


Fig. 1. Solution space for PD-SOI combinational circuits, showing existing tradeoffs between speed, leakage, and area, ranging from an all-FB to an all-TB circuit.

The general solution space is illustrated in Fig. 1, with each point in the space representing a different blend of TB and FB transistors. At one end of the space, an all-FB implementation will offer the highest speed possible (smallest delay) and will have the smallest area, but will suffer from the

This research was supported in part by Micronet and by the Semiconductor Research Corporation (SRC ID:2000-TJ-755).

maximum amount of leakage current. At the other end of the solution space, an all-TB implementation will have the minimum amount of leakage, but will be much larger in area, and much slower. The shaded surface on top of the space represents the area of acceptable solutions, i.e. specific combinations of TB and FB transistors that result in a reduced amount of leakage current compared to an all-FB implementation, with no reduction in speed. One corner of that surface represents the optimal solution that we seek, and results in the largest decrease in total leakage. If however the area penalty at that point is too large, we may choose to limit the number of TB devices allowed, and seek the best solution within the imposed constraint.

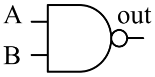
In Section II, we first present leakage-delay data that characterize basic logic gates. We then present in Section III an optimization methodology built around a specially developed static timing analysis (STA) algorithm that provides a solution for the best assignment of FB and TB devices in a large circuit. Finally, in Section IV, we present numerical results obtained by running our methodology on a number of ISCAS-85 benchmark circuits.

II. Gate-Level Characterization

To characterize a particular logic gate, we performed a number of simulations using Berkeley SPICE ver.3.4 with BSIM-PD-SOI ver.2.1 transistor models to obtain measurements of propagation delay and subthreshold leakage. For each gate, we performed separate measurements for each combination of FB/TB devices. As such, 16 different cases were considered for 2-input NAND and NOR gates, and 4 cases were considered for inverters. Whenever an NMOS TB transistor is used, we assume that a permanent connection is made between the transistor's body contact and ground. Similarly, for a PMOS TB device, a connection between that transistor's body contact and the positive voltage supply is established.

The delay characteristics of SOI logic gates have been analyzed by many authors, e.g. [3], [4], and it is well known that the gate delay strongly depends on the gate's initial logic state as well as its signal history. For circuit design purposes, especially when there is no prior knowledge of the eventual input signals, worst-case delays must be taken into consideration. We used a variety of switching patterns to capture the worst-case propagation delays, which typically occur when the gate has been at DC for a long time, or when the signals are switching at a low frequency. We measured two delay values for each input-to-output arc, namely the propagation delay when the output is rising, and the propagation delay when the output is falling. We refer to these quantities simply as *rising* and *falling* delays respectively throughout this paper. Compared to an all-FB logic gate, 'tying' a PMOS transistor increases the gate's rising delay, whereas tying an NMOS transistor increases the falling delay.

TABLE I
Average Leakage/Worst-Case Delay for a 2-Input NAND Gate

		PMOS						
		FB/FB	B tied	A tied	TB/TB	A –> out	B –> out	
NMOS	FB/FB	*247*	*178*	*178*	*108*	193	198	Falling delay
	B tied	*209*	*139*	*139*	*70*	203	221	
	A tied	*158*	*88*	*88*	*19*	229	232	
	TB/TB	*145*	*76*	*76*	*6*	239	262	
A –> out		116	116	132	132			
B –> out		130	151	130	151			
		Rising delay						

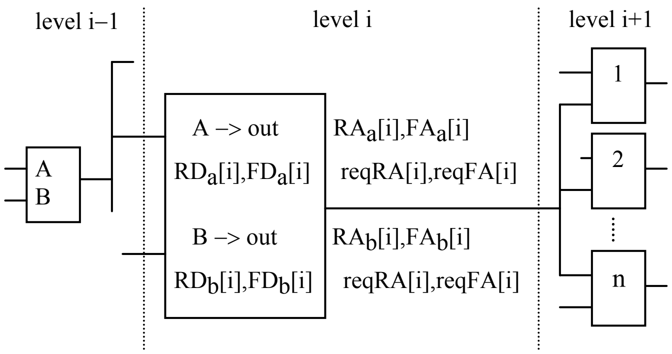
leakage in *pA*
delay in *ps*

Simulation results obtained for delay and leakage in a 2-input NAND gate are given in Table I. The leakage figures shown represent average leakage values calculated with an equal probability of input states, since the exact amount of leakage current in a gate depends on the actual input signals applied, which are not known. The average values shown are useful as a comparative measure between different FB/TB implementations. Leakage currents are given in the top-left part of the Table, and are in *pico-amperes*. Rising and falling delays are given in the bottom rows and the rightmost columns respectively. Separate delay values for each input-to-output arc are shown, with all delays in *pico-seconds*.

III. STA-Based Optimization

To obtain an FB/TB partition for a gate-level combinational circuit, we start off with an all-FB realization, and perform a block-oriented STA run to determine the critical path [5]. All the transistors contributing to the critical path are locked as FB-transistors. We then select one of the remaining FB devices and replace it with a TB device, and perform a new STA run to calculate the new delays. This process is repeated until there are no more devices that can be tied without increasing the delay of the circuit beyond that of the critical path, or until the imposed limit on the number of TB devices allowed has been reached. The resulting circuit has the same delay as the original all-FB circuit, however the total leakage current is dramatically reduced.

Each STA run consists of a forward propagation of signal arrival times, followed by a backward propagation of required arrival times. We developed an algorithm that traces separately all rising and falling delays in each gate, for each input-to-output arc. Delays associated with each gate (or block) in the circuit are defined in Fig. 2.



Key:
RD = delay if output is rising
FD = delay if output is falling
RA = actual rising arrival time
FA = actual falling arrival time
reqRA = required rising arrival time
reqFA = required falling arrival time

Fig. 2. Illustration of the block-oriented STA, with a key defining the various delay values associated with each block (gate) in the circuit.

All the primary inputs are assumed present at time t = 0, and the latest arriving signals are propagated one level at a time from the inputs to the output. The following equations are used in this procedure to calculate the actual arrival times from input *A* to the output of a gate in level *i*:

$$RA_a[i] = \mathrm{MAX}(FA_a[i\text{-}1]\,,\, FA_b[i\text{-}1]) + RD_a[i]$$

$$FA_a[i] = \mathrm{MAX}(RA_a[i\text{-}1]\,,\, RA_b[i\text{-}1]) + FD_a[i]$$

Similar equations are used to compute the arrival times propagated from input *B* to the output of a gate ($RA_b[i]$ and $FA_b[i]$). Once the last level is processed, the latest arrival time at the outputs is found, and is set as the required arrival time for all output signals at that level. This value is then back propagated through the circuit from the last level to the first to compute the required arrival times at the output of each block using the equations:

$$ReqRA[i] = \mathrm{MIN}(reqFA_j[i+1] - FD_j[i+1]), \quad j = 1 \text{ to } n$$

$$ReqFA[i] = \mathrm{MIN}(reqRA_k[i+1] - RD_k[i+1]), \quad k = 1 \text{ to } n$$

The difference between the required arrival time and the actual arrival time constitutes a *slack*. The critical path in the circuit is found by tracing all nodes with zero slack. Fig. 3 shows an actual STA run for the ISCAS-85 benchmark circuit *c17*, and shows the critical path with a delay of 526 *ps*. The delay values shown are taken from the data in Table I, and reflect an all-FB implementation.

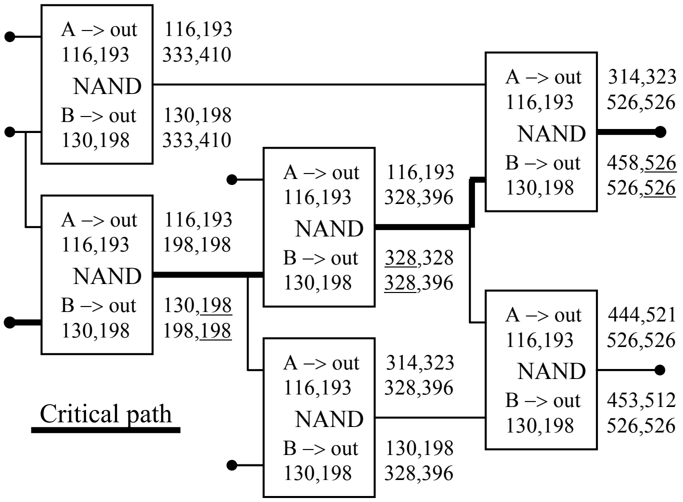


Fig. 3. Block-oriented static timing analysis run performed on the ISCAS-85 benchmark circuit *c17*. The critical path is represented by a thick line. All times are in *ps*.

By keeping rising and falling times separate, as well as by tracing each input-to-output arc individually, we can tell which transistor in the gate contributes to the critical path, and which one doesn't. A non-zero slack for a rising delay indicates that a particular PMOS transistor is off the critical path, and a TB device can therefore be used. Similarly, a non-zero slack for a falling delay indicates that an NMOS transistor is a candidate for tying. The available slack must obviously be large enough to allow an FB-TB substitution, otherwise the resulting change may result in increasing the delay beyond that of the original critical path. Typically, there are many FB-transistors in the circuit that are potential candidates for tying. In this work we follow a greedy approach by selecting each time a transistor that will result in the largest immediate gain if replaced by a TB transistor (we define `gain' as the ratio between the reduction in leakage current and the increase in delay, as calculated from the leakage/delay tables). After a TB device is introduced, a new STA run must be performed to re-compute all slacks before selecting a new candidate FB transistor for tying.

The basic block-oriented STA algorithm has a running time proportional to the number *n* of logic gates in the circuit. Assuming that each gate has *m* transistors, the number of STA runs needed to obtain an FB/TB partition is limited to a maximum of $n \times m$. The total running time of this methodology is therefore proportional to n^2, which is acceptable even for large circuits. This approach is simple, and guarantees a good FB/TB partition within the acceptable solution space, i.e. with no increase in the delay of the circuit. It doesn't however guarantee an optimal solution.

IV. Results

The above methodology was applied on a number of ISCAS-85 benchmark circuits. We first pre-processed the netlists to replace more complex gates with a combination of 2-input NAND, 2-input NOR, and NOT gates. We applied our methodology on each circuit twice, once without imposing any area constraints, and once with a limit of 30% on the percentage of TB devices allowed. The results obtained are shown in Table II. With an unconstrained area increase, the reduction in leakage achieved is between 51% and 86%. The resulting number and percentage of TB transistors are given for each circuit. With a constrained area increase, a maximum reduction in leakage of 51% was achieved, which is still very significant. For each circuit, we also followed two different approaches in choosing a transistor for tying after each STA run, whenever several transistors were found with the same potential leakage/delay gain. In one case we consistently picked the transistor that is the closest to the inputs; in the other case we chose the one closest to the outputs. The first approach worked better for some circuits, while the second approach worked better for the others. The figures shown in Table II reflect the best results obtained.

TABLE II
Optimization Results for some ISCAS-85 Benchmark Circuits

		Unconstrained area			Constrained area		
Circuit name	Total no. of transistors	TB transistors		Δ leakage	TB transistors		Δ leakage
		No.	%		No.	%	
c17	24	14	58%	–59%	7	29%	–33%
c432	1032	845	82%	–73%	309	30%	–36%
c499	2288	1298	57%	–51%	686	30%	–32%
c880	1878	1768	94%	–75%	563	30%	–40%
c1355	2352	1198	51%	–52%	705	30%	–34%
c2670	5660	5452	96%	–71%	1698	30%	–35%
c6288	10112	7988	79%	–86%	3033	30%	–51%

References

[1] Z. J. Lemnios, “Manufacturing technology challenges for low power electronics,” in *Dig. Tech. Papers, Symp. VLSI Technology,* 1995, pp. 5–8.

[2] G. G. Shahidi *et al.*, “SOI for a 1-Volt CMOS technology and applications to a 512-Kb SRAM with 3.5 ns access time,” in *IEDM Tech. Dig.,* 1993, pp. 813–816.

[3] Ruchir Puri and C. T. Chuang, “SOI digital circuits: design issues,” *IEEE Intl. Conference on VLSI Design*, pp. 474–479, 2000.

[4] C. T. Chuang *et al.*, SOI for digital CMOS VLSI: design considerations and advances,” *Proc. IEEE*, vol. 86, no. 4, pp. 689–720, 1998.

[5] R. B. Hitchcock, Sr., G. L. Smith, and D. D. Cheng, “Timing analysis of computer hardware”, *IBM J. Res. Develop.*, vol.26, no.1, pp. 100–105, 1982.