Industrial Engineering Journal
ISSN: 0970-2555
Volume : 51, Issue 8, August : 2022

AHL BASE ADAPTIVE HOLDING TECHNIQUE IMPLEMENTATION IN MULTIPLIER FOR DSP APPLICATIONS

[1]M.Santosh Reddy, [2]Dr.K.V.Ramparasad, [3]Dr.J.Kalippan, [4]Yagalla Haripal
[2,3]Professor,[1]Assistant Professor, [4]UG Student, [1,2,3,4]Dept. of Electronics and Communication Engineering,
Visvesvaraya College of Engineering and Technology, Mangalpalle, Telangana, India.

ABSTRACT
The speed of a processor is mostly determined by the circuit modules known as adders and subtractors. The sequential borrow bit propagation from LSB to MSB, which in turn depends on the number of bits of operands, limits the processing speed of subtraction (subtrahend and minuend). This research proposes two improved borrow select subtractor designs with higher area efficiency and reduced power consumption. The alterations made to the logical flow of subtraction by employing blocks with fewer logic gates result in fewer gates, which reduces the number of devices, the area, and the power consumption.
Keywords: Borrow Select Subtractor, Low power Subtractor, Area Efficient Subtractor, BSLS, Adder-Subtractor

INTRODUCTION

More and more complex signal processing systems are being built on a VLSI chip as the scale of integration keeps expanding. These signal processing applications need a lot of computing power, but they also use a lot of energy. Power consumption has emerged as a crucial issue in the design of modern VLSI systems, even if performance and area continue to be the two fundamental design constraints [There are two key drivers driving the need for low-power VLSI systems. Secondly, as operating frequency and processing capacity per chip continue to increase, big currents must be provided and the heat from high power consumption must be dissipated using appropriate cooling techniques. The second issue is the short battery life of portable electronics. Low power design directly leads to prolonged operation time in these portable devices. Subtraction usually impacts widely the overall performance of digital systems and a crucial arithmetic function. In electronic applications subtractors are most widely used. Applications where these are used are multipliers, DSP to execute various algorithms like FFT, FIR and IIR. Wherever concept of multiplication comes subtractors come in to the picture. As we know millions of instructions per second are performed in microprocessors. So, speed of operation is the most important constraint to be considered while designing multipliers. Due to device portability miniaturization of device should be high and power consumption should be low. Devices like Mobile, Laptops etc. require more battery backup. So, a VLSI designer has to optimize these three parameters in a design. These constraints are very difficult to achieve so depending on demand or application some compromise between constraints has to be made. Ripple carry subtractors exhibits the most compact design but the slowest in speed. Whereas carry look ahead is the fastest one but consumes more area. Carry select subtractors act as a compromise between the two subtractors. In 2002, a new concept of hybrid subtractors is presented to speed up subtraction process by Wang et al. that gives hybrid carry look-ahead/carry select subtractors design. In 2008, low power multipliers based on new hybrid full subtractors is presented. DESIGN of area- and power-efficient high-speed data path logic systems are one of the most substantial areas of research in VLSI system design. In digital subtractors, the speed of subtraction is limited by the time required to propagate a carry through the subtractor. The sum for each bit position in an elementary subtractor is generated sequentially only after the previous bit position has been summed and a carry propagated into the next position.

The CSLA is used in many computational systems to alleviate the problem of carry propagation delay by

independently generating multiple carries and then select a carry to generate the sum.

TYPES OF SUBTRACTORS

Subtraction is the most common and often used arithmetic operation on microprocessor, digital signal processor, especially digital computers. Also, it serves as a building block for synthesis all other arithmetic operations. Therefore, regarding the efficient implementation of an arithmetic unit, the binary subtractor structures become a very critical hardware unit. In any bookon computer arithmetic, someone looks that there exists a large number of different circuit architectures with different performance characteristics and widely used in the practice. Although many researches dealing with the binary subtractor structures have been done, the studies based on their comparative performance analysis are only a few. This is about a digital circuit. For an electronic circuit that handles analog signalssee mixer. In electronics, and subtractor or differentiator is a digital circuit that performs subtraction of numbers. In many computers and other kinds of processors, subtractors are used not only in the arithmetic logic unit(s), but also in other parts of the processor, where they are used to calculate addresses, table indices, and similar. Although subtractors can be constructed for many numerical representations, suchas binary-coded decimal or excess-3, the most common subtractors operate on binary numbers.In cases where two's complement or ones' complement is being used to represent negative numbers, it is trivial to modify an subtractor into an subtractor–subtractor. Other signed number representations require a more complex subtractor.

EXISTING MODEL

Adders, subtractors and multipliers are the essential building blocks of processors [1] [2] [3] [4] [5] [6]. Normally, subtraction is done using adder-subtractor module, which can practically result in slowing down arithmetic operation, since the same hardware has to be used for both addition and subtraction processes using additional control signals. The commonly used Ripple Borrow Subtractor (RBS) used for subtraction of unsigned numbers possesses a simple architecture. However, performance of RBS is limited by borrow propagation time incurred from Least Significant Bit (LSB) to Most Significant Bit (MSB). In other words, the delay of RBS depends on binary word length. Borrow Select Subtractor (BSLS) architecture is proposed here to overcome the limitations of existing RBS with multiple RBS circuits [1]. This method generates partial difference and borrow using multiplexers (MUX) and the final difference and borrow is selected. As the BSLS utilizes multiple RBS circuits in it, it is proved normally area inefficient. The conventional method of subtraction for signed numbers uses two’s complement method using addition. For addition process, references [7] [8] [9] [10] [11] [12] and [13] use adders. Note that [12] has lower power operational advantage and power delay product (PDP). Here, PDP is the product of power and delay. The proposed architectures are compared with the conventional two's complement method found in literature, basically employing an adder [14] [15]. The adder given in [12] has been considered for comparison of two’s complement method against the proposed architectures. For subtraction of signed numbers, a variant of BSLS [1] is proposed. The primary focus is on reducing area and power consumption, which is achieved by using logic blocks with fewer gates occupying less area even while aiming for same logic functionality.

OVERVIEW OF BSLS

Fig. 1 shows the structure of an half subtractor, implemented using an XOR gate, a NOT gateand an AND gate. While producing the output, half subtractor structure incurs a maximum latency of 2. Difference D incurs a latency of 1 and generation of borrow B incurs a latency of 2. Fig. 2 shows half adder implemented using an XOR gate and an AND gate. In order to produce sum S and carry C, this half adder incurs a maximum of one latency. Fig. 3 shows internal structure of a full subtractor realized using 2 XOR gates, 2 NOT gates, 2 AND

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 51, Issue 8, August : 2022

gates and 1 OR gate. The full subtractor incurs a maximum latency of 4. Realization of difference D incurs a latency of 2 and borrow B introduces a latency of 4.

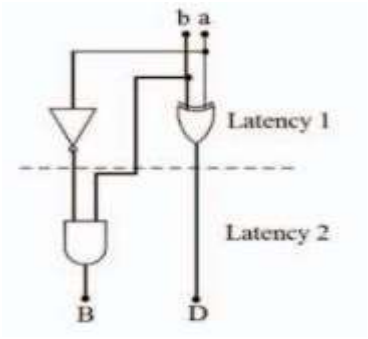


Fig. 1. Half Subtractor [1]

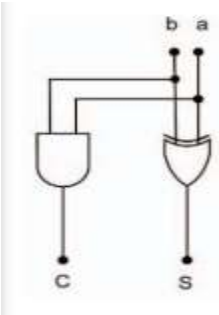


Fig. 2. Half Adder [1]

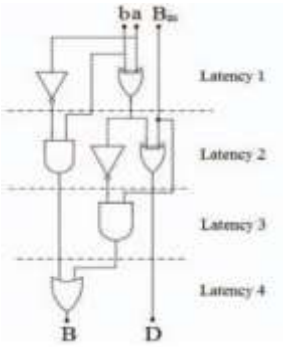


Fig. 3. Full Subtractor Gate Structure [1]

Fig. 4 depicts the structure of 2-bit Binary to Excess-1 Converter (BEC) implemented using 1 XOR, 1 AND and 1 NOT gate. The 16-bit BSLS for unsigned numbers consists of five logic groups. Each group computes the difference and borrow for different number of bits, as shown inTable I. Each functional group except first group has 2 n-bit RBS, n-bit BEC and a (2n+2):(n+1) multiplexer. The first group has only one two-bit RBS. The minuend and subtrahend are the inputs to RBS. This accounts to generation of the difference bit output when the borrow from theprevious stage is 0. The minuend is increased by one using the BEC block. The subtrahend and increased minuend is given as input to RBS. This accounts to generation of difference bit when the borrow from previous stage is 1. The difference and borrow bits are available for borrow as 0and 1. The correct output is selected based on the borrow bit obtained from the previous stageusing the MUX.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 51, Issue 8, August : 2022

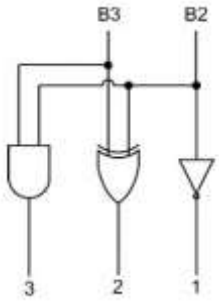


Fig. 4. 2-bit BEC [1]

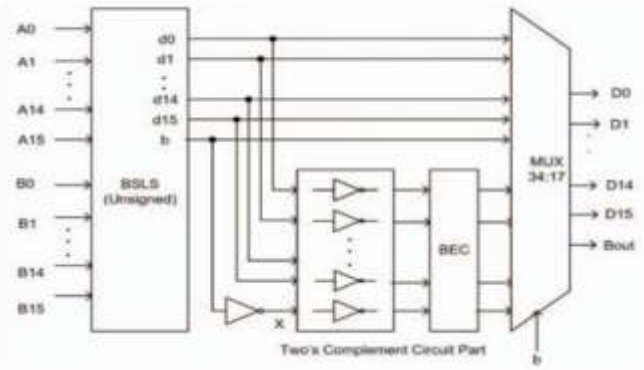


Fig. 5. Signed Borrow Select Subtractor [1]

The difference between the subtrahend and the minuend is in 2's complement form when the subtrahend is greater than minuend. To convert the two's complement form into signed magnitude form, additional blocks are required. When borrow output from unsigned BSLS is 0, the result is positive. Hence, the output doesn't necessitate the conversion to signed magnitude form. Similarly, as borrow output is 1, the result is negative. Then, the output of borrow-out is complemented and 2's complement is performed on entire result in signed magnitude form. Fig.5 depicts the signed borrow select subtractor.

PROPOSED METHOD

Razor approach

Razor, a new approach to DVS, is based on dynamic detection and correction of speedpath failures in digital designs. Its key idea is to tune the supply voltage by monitoring the error rate during operation. Because this error detection provides in-place monitoring of the actual circuit delay, it accounts for both global and local delay variations and doesn't suffer from voltage scaling disparities. It therefore eliminates the need for voltage margins to ensure always- correct circuit operation in traditional designs. In addition, a key Razor feature is that operation at subcritical supply voltages doesn't constitute a catastrophic failure but instead represents a trade-off between the power penalty incurred from error correction and the additional power savings obtained from operating at a lower supply voltage. The Razor project includes four faculty members and more than 10 graduate students in the Electrical Engineering and ComputerScience Department at the University of Michigan. We implemented the Razor technique in a prototype 64-bit Alpha processor design and used it to obtain a realistic measure of the power savings and overhead for inplace error detection and correction. We also studied the error rate trends for data path components, using both circuit-level simulation and silicon measurements of a fullcustom multiplier block. Architectural simulations then helped us analyze the overall throughput and power characteristics of Razor-based DVS for different benchmark test programs. On average, Razor reduced

simulated consumption by more than 40 percent, compared with traditional design-time DVS and delay-chain-based approaches.

Razor error detection and correction

Razor relies on a combination of architectural and circuit-level techniques for efficient error detection and correction of delay path failures. Figure 3.1 illustrates the concept for a pipeline stage. A so-called shadow latch, controlled by a delayed clock, augments each flip- flop in the design.

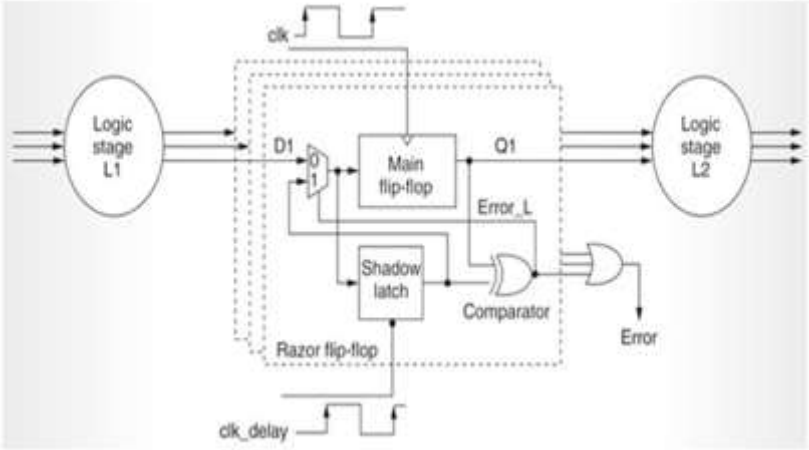


Fig. 6.Pipeline stage augmented with Razor latches and control lines.

In a given clock cycle, if the combinational logic, stage L1, meets the setup time for the main flip-flop for the clock's rising edge, then both the main flip-flop and the shadow latch will latch the correct data. In this case, the error signal at the XOR gate's output remains low, leaving the pipeline's operation unaltered. If combinational logic L1 doesn't complete its computation in time, flip-flop will latch an incorrect value, while the shadow latch will latch the late-arriving correct value. The error signal would then go high, prompting restoration of the correct value from the shadow latch into the main flip-flop, and the correct value becomes available to stage L2. To guarantee that the shadow latch will always latch the input data correctly, designers constrain the allowable operating voltage so that under worst-case conditions the logic delay doesn't exceed the shadow latch's setup time. If an error occurs in stage L1 during a particular clock cycle, the data in L2 during the following clock cycle is incorrect and must be flushed fromthe pipeline. However, because the shadow latch contains the correct output data from stage L1, the instruction needn't re execute through this failing stage. Thus, a key Razor feature is that if an instruction fails in a particular pipeline stage, it re executes through the following pipeline stage while incurring a one-cycle penalty. The proposed approach therefore guarantees a failing instruction's forward progress, which is essential to avoid perpetual failure of an instruction at a particular pipeline stage. We limit this article to Razor's use on combinational logic blocks contained within the pipeline data paths. Therefore, we apply Razor to a simple the processor that uses an in-order pipeline with simple control and small caches. In such a processor, control logic and SRAM structures remain error free, even at the worst-case frequency and voltage. Therefore, they don't require Razor technology.

Circuit-level implementation issues

Razor-based DVS requires that the error detection and correction circuitry's delay and power overhead remain minimal during errorfree operation. Otherwise, this circuitry's power overhead would cancel out the power savings from more-aggressive voltage scaling. In addition, it's necessary to minimize error correction overhead to enable efficient operation at moderate error rates. We applied several methods to reduce the Razor flip-flop's power and delay overhead, as shown in Figure 3.2.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 51, Issue 8, August : 2022

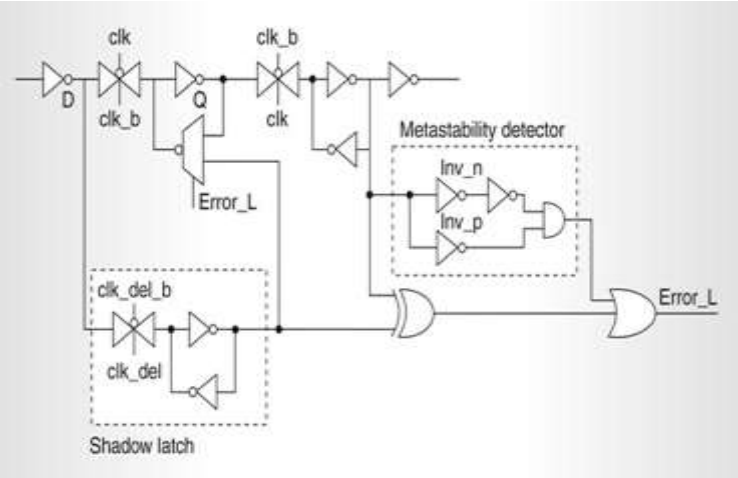


Fig .7.Reduced-overhead Razor flip-flop and met stability detection circuits.

The multiplexer at the Razor flip-flop's input causes a significant delay and poweroverhead therefore, we moved it to the feedback path of the main flip- flop's master latch, as Figure 3.2 shows. Hence, the Razor flip-flop introduces only a slight increase in the critical path's capacitive loading and has minimal impact on the design's performance and power. In most cycles, a flip-flop's input will not transition, and the circuit will incur only the power overhead from switching the delayed clock, thereby reducing Razor's power overhead. Generating the delayed clock locally reduces its routing capacitance, which further minimizes additional clock power. Simply inverting the main clock will result in a clock delayed by half the clock cycle. Also, many noncritical flip-flops in the design don't need Razor. If the maximum Main flip-flop Shadow latch Logic stage L2 Logic stage L1 D1 clkclk_delay 0 1 Comparator Error Error_L Q1 Razor flip-flop Figure 1. Pipeline stage augmented with Razor latches and control lines. delay at a flip-flop's input is guaranteed to meet the required cycle time under the worst-case subcritical voltage, the flip-flop cannot fail and doesn't need replacement with a Razor flip-flop. We found that in the prototype Alpha processor, only 192 flip-flops out of 2,408 required Razor, which significantly reduced the Razor approach's power overhead. For this prototype processor, the total simulated power overhead in error-free operation (owing to Razor flip- flops) was less than 1 percent, while the delay overhead was negligible. Using a delayed clock at the shadow latch raises the possibility that a short path in the combinational logic will corrupt the data in the shadow latch. To prevent corruption of the shadow latch data by the next cycle's data, designers add a minimum-path-length constraint at each Razor flip-flop's input. These minimum-path constraints result in the addition of buffers during logic synthesis to slow down fast paths; therefore, they introduce a certain power overhead. The minimum-path constraint is equal to clock delay tdelay plus the shadow latch's hold time, thold. A large clock delay increases the severity of the short-path constraint and therefore increases the power overhead resulting from the need for additional buffers. On the other hand, a small clock delay reduces the margin between the main flip-flop and the shadow latch, hence reducing the amount by which designers can drop the supply voltage below the critical supply voltage. In the prototype 64-bit Alpha design, the clock delay was half the clock period. This simpli- fied generation of the delayed clock while continuing to meet the short-path constraints, resulting in asimulated power overhead (because of buffers) of less than 3 percent. The adaptive logic program aims at developing a type of formal logics (and the connected theory) that is especially suited to explicate the many interesting dynamic consequence relations that occur in human reasoning but for which there is no positive test (see the next section). Such consequence relations occur, for example, in inductive reasoning, handling inconsistent data.The explication of such consequence relations is realized by the dynamic proof theories of adaptive logics. These proof theories are dynamic in that formulas

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 51, Issue 8, August : 2022

derived at some stage may not be derived at a later stage, and vice versa.This program is application driven. This is one of the reasons why the predicative level is considered extremely important, even if, for many adaptive logics, the basic features of the dynamics are already present at the propositional level. The main applications are taken from the philosophy of science; some also from more pedestrian contexts.The interest in dynamic consequence relations led rather naturally to an interest in dynamic aspects of reasoning. Some of these already occur in Classical Logic.

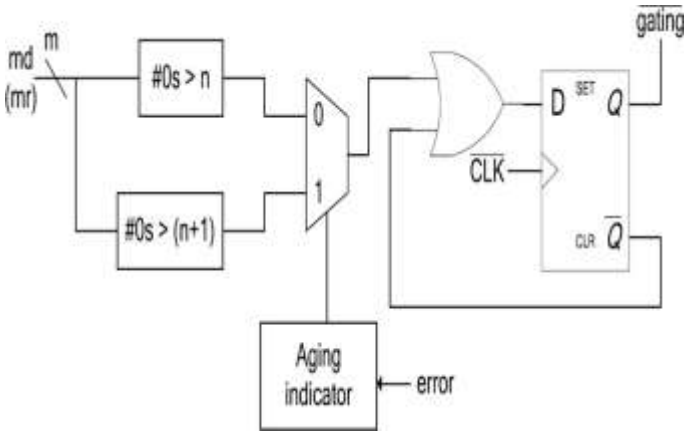
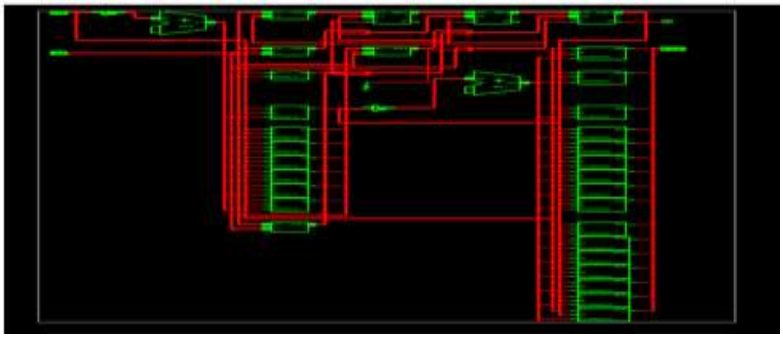


Fig .8. Diagram of AHL (md means multiplicand; mr means multiplicator).

SIMULATION RESULTS

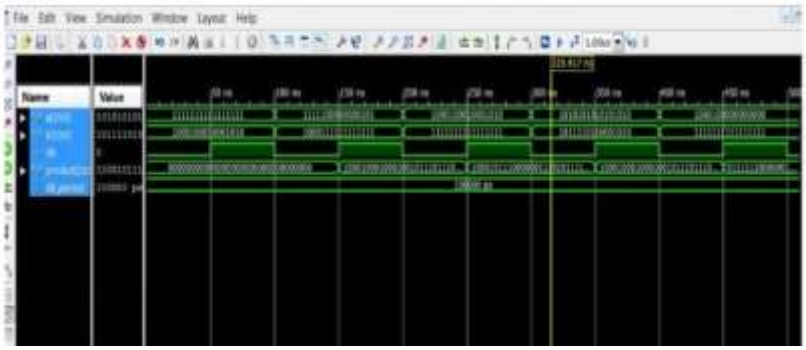
RTL

INTERNAL BLOCK DIAGARAM



SIMULATION OUTPUT WAVEFORMS

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 51, Issue 8, August : 2022



CONCLUSION

The conclusion states the proposed design of an aging-aware variable-latency multiplier design with the AHL. The multiplier is able to adjust the AHL to mitigate performance degradation due to increased delay. The experimental results show that our proposed architecture from 4bit to 16bit multiplication with Booth as last stage instead of Normal RCA adder it will decrease the delay and improve the performance compared with previous designs. In the proposed architecture instead of the column- and row-bypassing multipliers the booth multiplier can be examined by the number of zeros in either the multiplicand or multiplication to predict whether the operation requires one cycle or two cycles to complete.When input patterns are random, the number of zeros and ones in the multiplication and multiplicand follows a normal distribution. Therefore, using the number of zeros or ones as the judging criteria results in similar outcomes. Hence, the two aging-aware multipliers can be implemented using similar architecture, and the difference between the two bypassing multiplierslies in the input signals of the AHL.

Future Scope

The technique called Recovery Boosting that allows both pMOS circuit to be put into the recovery mode by slightly modifying to the design. Thus this helps us in Increasing the speed, Reduction in complexity of design, Decreasing the instability effects and providing 100% accurate results.

REFERENCES

1. Y. Cao. (2013). Predictive Technology Model (PTM) and NBTI Model
2. S. Zafaret al., “A comparative study of NBTI and PBTI (charge trapping) in SiO2/HfO2 stacks with FUSI, TiN, Re gates,” in Proc.IEEESymp. VLSI Technol. Dig. Tech. Papers, 2006, pp. 23–25.
3. S. Zafar, A. Kumar, E. Gusev, and E. Cartier, “Thresh-old voltage instabilities in high-k gate dielectric stacks,” IEEE Trans. Device Mater.Rel., vol. 5, no. 1, pp. 45–64, Mar. 2005.
4. H.-I. Yang, S.-C.Yang, W. Hwang, and C.-T. Chuang, “Impacts of NBTI/PBTI on timing control circuits and degradation tolerant designinnanoscale CMOS SRAM,” IEEE Trans. Circuit Syst., vol. 58, no. 6, pp. 1239–1251, Jun. 2011.
5. R. Vattikonda, W. Wang, and Y. Cao, “Modeling and miimization of pMOS NBTI effect for robust naometer design,” in Proc. ACM/IEEEDAC, Jun. 2004, pp. 1047– 1052.
6. H. Abrishami, S. Hatami, B. Amelifard, and M. Pe-dram, “NBTI-aware flip-flop characterization and de-sign,” in Proc. 44th ACM GLSVLSI,2008, pp. 29–34
7. S. V. Kumar, C. H. Kim, and S. S. Sapatnekar, “NBTI-aware synthesis of digital circuits,” in Proc. ACM/IEEE DAC, Jun. 2007, pp. 370–375.
8. A. Calimera, E. Macii, and M. Poncino, “Design tech-niqures for NBTItolerant power-gating architecture,” IEEE Trans. Circuits Syst., Exp.Briefs, vol. 59, no. 4, pp. 249–253, Apr. 2012.

9. K.-C. Wu and D. Marculescu, “Joint logic restructur-ing and pin reordering against NBTI-induced perfor-mance degradation,” in Proc. DATE,2009, pp. 75–80.
10. Y. Lee “A fine-grained technique of NBTI-aware voltage scaling and body biasing for stan-dard cell based designs,” in Proc. ASPDAC,2011, pp. 603–608.