Literature Review on Aging-Aware Reliable Multiplier Design with Adaptive Hold Logic

Chanchal Singh[1], Dr. Mukul Shrivastava[2], Dr. Rita Jain[3]

[1]M-Tech Research Scholar,[2] Research Guide, [3]HOD, Deptt. of Electronics & Communication

LNCT, Bhopal

Abstract: Shrinking feature sizes in CMOS-based technology beyond the 45nm regime have given rise to increased levels of variation in digital circuits and architectures due to process, temperature, and aging effects. The fabrication process induces variations in the process parameters, causing differing levels of perturbation in the circuit delay in each manufactured part at the post silicon stage. Moreover, after manufacturing, during the normal operation of a chip, new variations are injected due to various aging mechanisms, particularly Bias Temperature Instability (BTI). These effects cause long-term degradations in transistor performance, resulting in temp oral delay degradations at the circuit level. The mechanism of BTI is exacerbated as transistor sizes reduce, and poses a growing threat to circuit reliability.

Keywords— Adaptive hold logic (AHL), negative bias temperature instability (NBTI), positive bias temperature instability (PBTI), reliable multiplier, variable latency.

I. INTRODUCTION

Many DSP applications demand high throughput and real-time response, performance constraints that often dictate unique architectures with high levels of concurrency. DSP designers need the capability to manipulate and evaluate complex algorithms to extract the necessary level of concurrency. Performance constraints can also be addressed by applying alternative technologies. A change at the implementation level of design by the insertion of a new technology can often make viable an existing marginal algorithm or architecture.

The VHDL language supports these modeling needs at the algorithm or behavioral level, and at the implementation or structural level. It provides a versatile set of description facilities to model DSP circuits from the system level to the gate level. Recently, we have also noticed efforts to include circuit-level modeling in VHDL. At the system level we can build behavioral models to describe algorithms and architectures. We would use concurrent processes with constructs common to many high-level languages, such as if, case, loop, wait, and assert statements. VHDL also includes user-defined types, functions, procedures, and packages." In many respects VHDL is a very powerful, high-level, concurrent programming language. At the implementation level can build be structural models using component instantiation statements that connect and invoke subcomponents. The VHDL generate statement provides ease of block replication and control. A dataflow level of description offers a combination of the behavioral and structural levels of description. VHDL lets us use all three levels to describe a single component. Most importantly, the standardization of VHDL has spurred the development of model libraries and design and development tools at every level of abstraction. VHDL, as a consensus description language and design environment, offers design tool portability, easy technical exchange, and technology insertion.

VHDL

An entity declaration, or entity, combined with architecture or body constitutes a VHDL model. VHDL calls the entity-architecture pair a design entity. By describing alternative architectures for an entity, we can configure a VHDL model for a specific level of investigation. The entity contains the interface description common to the alternative architectures. It communicates with other entities and the environment through ports and generics. Generic information particularizes an entity by specifying environment constants such as register size or delay value. For example,

```
entity A is
        port (x, y: in real; z: out real);
        generic (delay: time);
end A;
```

The architecture contains declarative and statement sections. Declarations form the region before the reserved word begin and can declare local elements such as signals and components. Statements appear after *begin* and can contain concurrent statements. For instance,

```
architecture B of A is
        component M
        port ( j : in real ; k : out real);
```

```
        end component;
 signal a,b,creal := 0.0;
begin
        "concurrent statements"
end B;
```

The variety of concurrent statement types gives VHDL the descriptive power to create and combine models at the structural, dataflow, and behavioral levels into one simulation model. The structural type of description makes use of component instantiation statements to invoke models described elsewhere. After declaring components, use them in the component instantiation statement, assigning ports to local signals or other ports and giving values to generics. invert: M port map (j => a ; k => c); We can then bind the components to other design entities through configuration specifications in VHDL's architecture declarative section or through separate configuration declarations.

The dataflow style makes wide use of a number of types of concurrent signal assignment statements, which associate a target signal with an expression and a delay. The list of signals appearing in the expression is the sensitivity list; the expression must be evaluated for any change on any of these signals. The target signals obtain new values after the delay specified in the signal assignment statement. If no delay is specified, the signal assignment occurs during the next simulation cycle:

```
c <= a + b after delay;
```

VHDL also includes conditional and selected signal assignment statements. It uses block statements to group signal assignment statements and makes them synchronous with a guarded condition. Block statements can also contain ports and generics to provide more modularity in the descriptions. We commonly use concurrent process statements when we wish to describe hardware at the behavioral level of abstraction. The process statement consists of declarations and procedural types of statements that make up the sequential program. Wait and assert statements add to the descriptive power of the process statements for modeling concurrent actions:

```
process
begin
        variable i : real := 1.0;
wait on a;
        i = b * 3.0;
        c <= i after 1 ns;
end process;
```

Other concurrent statements include the concurrent assertion statement, concurrent procedure call, and generate statement. Packages are design units that permit types and objects to be shared. Arithmetic operations dominate the execution time of most Digital Signal Processing (DSP) algorithms and currently the time it takes to execute a multiplication operation is still the dominating factor in determining the instruction cycle time of a DSP chip and Reduced Instruction Set Computers (RISC). Among the many methods of implementing high speed parallel multipliers, there is one basic approach namely Booth algorithm.

II. SYSTEM MODEL

NBTI

Negative Bias Temperature Instability (NBTI) has emerged as a major reliability challenge for the semiconductor industry in recent years. NBTI impact is getting worse in each technology generation with greater performance and reliability loss. When a negative voltage is applied at a p channel transistor (PMOS) gate, interface traps are formed near oxide layer, causing a change in transistor characteristics. When the input to a PMOS is low (logic zero), the transistor is in a stress phase. During the stress phase, the transistor parameters slowly deviate from the nominal value. When the input to the PMOS is high (logic one), the transistor is in a recovery phase. During the recovery phase, trapped charges are released, regaining the original transistor state. The PMOS enters into stress and recovery phases alternately, when the input to the PMOS is dynamic. Longer the stress period, higher is the impact of NBTI on transistor parameters. Therefore, input to the transistor indirectly determines the extent of NBTI degradation.

MULTIPLIER

A Binary multiplier is an electronic hardware device used in digital electronics or a computer or other electronic device to perform rapid multiplication of two numbers in binary representation. It is built using binary adders.

The rules for binary multiplication can be stated as follows

1. If the multiplier digit is a 1, the multiplicand is simply copied down and
2. Represents the product.
3. If the multiplier digit is a 0 the product is also 0.

For designing a multiplier circuit we should have circuitry to provide or do the following three things:

1. It should be capable identifying whether a bit 0 or 1 is.

2. It should be capable of shifting left partial products.
3. It should be able to add all the partial products to give the products as sum of partial products.
4. It should examine the sign bits. If they are alike, the sign of the product will be a positive, if the sign bits are opposite product will be negative. The sign bit of the product stored with above criteria should be displayed along with the product.

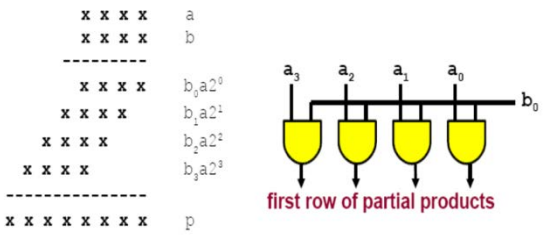
From the above discussion we observe that it is not necessary to wait until all the partial products have been formed before summing them. In fact the addition of partial product can be carried out as soon as the partial product is formed.

Binary multiplication (eg n=4)

p=a×b
an−1 an−2….a1a0
bn−1bn−2….b1b0
p2 n−1 p2 n−2….p1 p0

```
                 xxxx a
                 xxxx b
                 ---------
                      xxxx     b0a20
                      xxxx     b1a21
               xxxx            b2a22
     xxxx                      b3a23
                                ---------------
          Xxxxxxxx          p
```

Partical products
In the partial products are trivial-
If multiplier bit =1, copy the multiplicand
Else 0
Use an 'AND' gate!



III. LITERATURE SURVEY

C. Lin, Y. H. Cho and Y. M. Yang, [1] Digital multipliers are among the most critical arithmetic functional units. The overall performance of these systems depends on the throughput of the multiplier. Meanwhile, the negative bias temperature instability effect occurs when a pMOS transistor is under negative bias (Vgs = -Vdd), increasing the threshold voltage of the pMOS transistor, and reducing multiplier speed. In this research, we propose an aging-aware multiplier design with novel adaptive hold logic (AHL) circuit. The experimental results show that our proposed architecture with 16 × 16 and 32 × 32 column-bypassing multipliers can attain up to 62.88% and 76.28% performance improvement, respectively, compared with 16×16 and 32×32 fixed-latency column-bypassing multipliers. Furthermore, proposed architecture with 16 × 16 and 32 × 32 row-bypassing multipliers can achieve up to 80.17% and 69.40% performance improvement as compared with 16×16 and 32 × 32 fixed-latency row-bypassing multipliers.

S. Zafar et al.,[2] Threshold voltage (Vt) of a field effect transistor (FET) is observed to shift with stressing time and this stress induced V t shift is an important transistor reliability issue. Vt shifts that occur under negative gate bias is referred as NBTI and those that occur under positive bias is referred as PBTI or charge trapping. In this paper, present a comparative study of NBTI and PBTI for a variety of FETs with different dielectric stacks and gate materials. The study has two parts. In part I, NBTI and PBTI measurements are performed for FUSI NiSi gated FETs with SiO2 SiO2/HfO2 and SiO2/HfSiO as gate dielectric stacks and the results are compared with those for conventional SiON/poly-Si FETs. The main results are: (i) NBTI for SiO 2/NiSi and SiO2/HfO2/NiSi are same as those conventional SiON/poly-Si FETs; (ii) PBTI significantly increases as the Hf content in the high K layer is increased; and (iii) PBTI is a greater reliability issue than NBTI for HfO2/NiSi FETs. In part II of the study, NBTI and PBTI measurements are performed for SiO2/HfO2 devices with TiN and Re as gates and the results are compared with those for NiSi gated FETs. The main results are: (i) NBTI for SiO 2/HfO2/TiN and SiO2/HfO2/Re pFETs are similar with those observed for NiSi gated pFETs; and (ii) PBTI in TiN and Re gated HfO2 devices is much smaller than those observed for SiO2/HfO2/NiSi. In summary for SiO2/HfO2 stacks, NBTI is observed to be independent of gate material whereas PBTI is significantly worse for FUSI gated devices. Consequently, HfO2 FETs with TiN and Re gates exhibit over all superior transistor reliability characteristics in comparison to HfO2/FUSI FETs.

S. Zafar, A. Kumar, E. Gusev and E. Cartier,[3] Over recent years, there has been increasing research and development efforts to replace SiO2 with high dielectric constant (high-κ) materials such as HfO2, HfSiO, and Al2O3. An important transistor reliability issue is the threshold voltage stability under prolonged stressing. In these materials, threshold voltage is observed to shift with stressing time and conditions, thereby giving rise to threshold voltage instabilities. In this paper, review various causes of threshold voltage instability: charge

trapping under positive bias stressing, positive charge creation under negative bias stressing (NBTI), hot-carrier stressing, de-trapping and transient charge trapping effects in high-κ gate dielectric stacks. Experimental and modeling studies for these threshold voltage instabilities are reviewed.

H. I. Yang, S. C. Yang, W. Hwang and C. T. Chuang,[4] Negative-bias temperature instability (NBTI) and positive-bias temperature instability (PBTI) weaken PFET and NFET over the lifetime of usage, leading to performance and reliability degradation of nanoscale CMOS SRAM. In addition, most of the state-of-the-art SRAM designs employ replica timing control circuit to mitigate the effects of leakage and process variation, optimize the performance, and reduce power consumption. NBTI and PBTI also degrade the timing control circuits and may render them ineffective. In this paper, author provide comprehensive analyses on the impacts of NBTI and PBTI on a two-port 8T SRAM design, including the stability and Write margin of the cell, Read/Write access paths, and replica timing control circuits. Show, for the first time, that because the Read/Write replica timing control circuits are activated in every Read/Write cycle, they exhibit distinctively different degradation behavior from the normal array access paths, resulting in degradation of timing control and performance. Also discuss degradation tolerant design techniques to mitigate the performance and reliability degradation induced by NBTI/PBTI.

R. Vattikonda, Wenping Wang and Yu Cao, [5] Negative bias temperature instability (NBTI) has become the dominant reliability concern for nanoscale PMOS transistors. In this paper, a predictive model is developed for the degradation of NBTI in both static and dynamic operations. Model scalability and generality are comprehensively verified with experimental data over a wide range of process and bias conditions. By implementing the new model into SPICE for an industrial 90nm technology, key insights are obtained for the development of robust design solutions: (1) the most effective techniques to mitigate the NBTI degradation are VDD tuning, PMOS sizing, and reducing the duty cycle; (2) an optimal VDD exists to minimize the degradation of circuit performance; (3) tuning gate length or the switching frequency has little impact on the NBTI effect; (4) a new switching scenario is identified for worst case timing analysis during NBTI stress.

IV.PROBLEM IDENTIFICATION

This paper proposed 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 the proposed architecture with 16× 16 and 32× 32 column-bypassing multipliers can attain up to 62.88% and 76.28% performance improvement compared with the 16 × 16 and 32 × 32 FLCB multipliers, respectively. Electro-migration occurs when the current density is high enough to cause the drift of metal ions along the direction of electron flow. The metal atoms will be gradually displaced after a period of time, and the geometry of the wires will change. If a wire becomes narrower, the resistance and delay of the wire will be increased, and in the end, Electromigration may lead to open circuits. This issue is also more serious in advanced process technology because metal wires are narrower, and changes in the wire width will cause larger resistance differences.

V. CONCLUSION

Multipliers are one the most important component of many systems. So always need to find a better solution in case of multipliers. Multipliers should always consume less power and cover less power. So through this review paper we try to determine which of the algorithm works the best. If the aging effects caused by the BTI effect and electro-migration are considered together, the delay and performance degradation will be more significant. Fortunately, variable latency multipliers can be used under the influence of both the BTI effect and electro-migration. In addition, variable latency multipliers have less performance degradation because variable latency multipliers have less timing waste, but traditional multipliers need to consider the degradation caused by both the BTI effect and electro-migration and use the worst case delay as the cycle period.

REFERENCES

[1] I. C. Lin, Y. H. Cho and Y. M. Yang, "Aging-Aware Reliable Multiplier Design With Adaptive Hold Logic," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 23, no. 3, pp. 544-556, March 2015.

[2] S. Zafar et al., "A Comparative Study of NBTI and PBTI (Charge Trapping) in SiO2/HfO2 Stacks with FUSI, TiN, Re Gates," 2006 Symposium on VLSI Technology, 2006. Digest of Technical Papers., Honolulu, HI, 2006, pp. 23-25.

[3] S. Zafar, A. Kumar, E. Gusev and E. Cartier, "Threshold voltage instabilities in high-κ gate dielectric stacks," in IEEE Transactions on Device and Materials Reliability, vol. 5, no. 1, pp. 45-64, March 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 Design in Nanoscale CMOS SRAM," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 58, no. 6, pp. 1239-1251, June 2011.

[5] R. Vattikonda, Wenping Wang and Yu Cao, "Modeling and minimization of PMOS NBTI effect for robust nanometer design," 2006 43rd ACM/IEEE Design Automation Conference, San Francisco, CA, 2006, pp. 1047-1052.

[6] H. Abrishami, S. Hatami, B. Amelifard, and M. Pedram, “NBTI-aware flip-flop characterization and design,” 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 techniqures 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 restructuring and pin reordering against NBTI-induced performance degradation,” in Proc. DATE, 2009, pp. 75–80.

[10] Y. Lee and T. Kim, “A fine-grained technique of NBTI-aware voltage scaling and body biasing for standard cell based designs,” in Proc. ASPDAC, 2011, pp. 603–608.

[11] M. Basoglu, M. Orshansky, and M. Erez, “NBTI-aware DVFS: A new approach to saving energy and increasing processor lifetime,” in Proc. ACM/IEEE ISLPED, Aug. 2010, pp. 253–258.

[12] K.-C. Wu and D. Marculescu, “Aging-aware timing analysis and optimization considering path sensitization,” in Proc. DATE, 2011, pp. 1–6.

[13] K. Du, P. Varman, and K. Mohanram, “High performance reliable variable latency carry select addition,” in Proc. DATE, 2012, pp. 1257–1262.

[14] A. K. Verma, P. Brisk, and P. Ienne, “Variable latency speculative addition: A new paradigm for arithmetic circuit design,” in Proc. DATE, 2008, pp. 1250–1255.

[15] D. Baneres, J. Cortadella, and M. Kishinevsky, “Variable-latency design by function speculation,” in Proc. DATE, 2009, pp. 1704–1709.

[16] Y.-S. Su, D.-C. Wang, S.-C. Chang, and M. Marek-Sadowska, “Performance” optimization using variable-latency design style,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 19, no. 10, pp. 1874–1883, Oct. 2011.