Low Power Soft Error Tolerant Macro Synchronous Micro Asynchronous (MSMA)Pipeline

F. K. Lodhi [1], S. R. Hasan [2], O. Hasan [1] and F. Awwad [3]
[1] *Sch. of Elect. Engg. and Comp. Sc., National University of Sciences and Technology (NUST), Islamabad, Pakistan*
[2] *Department of Electrical and Computer Engineering, Tennessee Technological University, Cookeville, TN, USA*
[3] *College of Engineering, United Arab Emirates University, Al-Ain, UAE*
{faiq.khalid, osman.hasan}@seecs.edu.pk, shasan@tntech.edu, f_awwad@uaeu.ac.ae

Abstract — **Advancement in deep submicron (DSM) technologies led to miniaturization. However, it also increased the vulnerability against some electrical and device non-idealities, including the soft errors. These errors are significant threat to the reliable functionality of digital circuits. Several techniques for the detection and deterrence of soft errors (to improve the reliability) have been proposed, both in synchronous and asynchronous domain. In this paper we propose a low power and soft error tolerant solution for synchronous systems that leverages the asynchronous pipeline within a synchronous framework. We named our technique as macro synchronous micro asynchronous (MSMA) pipeline. We provided a framework along with timing analysis of the MSMA technique. MSMA is implemented using a macro synchronous system and soft error tolerant and low power version of null convention logic (NCL) asynchronous circuit. It is found out that this solution can easily replace the intermediate stages of synchronous and asynchronous pipelines without changing its interface protocol. Such NCL asynchronous circuits can be used as a standard cell in the synchronous ASIC design flow. Power and performance analysis is done using electrical simulations, which shows that this techniques consumes at least 22% less power and 45% less energy delay product (EDP) compared to state-of-the-art solutions.**

Keywords-component; Soft Error, NCL pipeline, SE tolrerant circuits, Low power Aynshronous circuits

I. INTRODUCTION

With the advancements in modern deep submicron (DSM) technologies, the semiconductor devices are becoming more sensitive to the soft errors (SE) [1]. These errors can lead to malfunctioning in digital systems. Different techniques have been devised to mitigate soft errors, which can be broadly categorized into redundancy based technique (e.g. triple modular redundancy (TMR)) or error detection and correction (EDAC) codes [2]. These conventional techniques are to mitigate soft errors in sequential designs and memory elements only. In modern DSM technologies combinational circuits are also subject to soft error propagation [2].

Vulnerability against soft errors in combinational circuits and lack of soft error protection for combinational logic poses a major challenge to future high performance mission critical computing. Due to inherent error detection capability of quasi delay insensitive (QDI) asynchronous circuits, researchers explored them for the prevention and detection of soft errors. In [3, 4] authors proposed QDI based asynchronous circuit using full duplication of circuits and synchronizing the replicated results through the C elements [5] to introduce soft error tolerance. Jang et al. [6] evaluated QDI asynchronous interfaces for single event upset (SEU) tolerance. Peng et al. [7] developed a detection method for concurrent failures in pipelined asynchronous circuits. Gardiner et al. proposed to use a stoppable latch that can prevent the propagation of the fault data [8]. These asynchronous techniques resolve SE from communication interface perspective, hence they cannot be used readily as an alternative to conventional combinational circuits.

In this paper we leveraged a recently proposed asynchronous pipeline techniques by Kuang et al. [9] and F. K. Lodhi et al. [10]. The authors in [9] proposed a technique to detect and correct the soft errors by using the dual rail property of the threshold gates in Null Conventional Logic (NCL) pipelines. In [11] the soft error tolerant threshold gates are proposed. In this paper we explored the conjunction of [9] and [11] to make a combinational circuit equivalent, this allows us to make any logic block into soft error tolerant asynchronous circuits. Unlike the methods proposed by Kuang [9] and F. K. Lodhi [10] this does not require any extra circuitry for soft error tolerance. The low power threshold gate proposed in [11] is also a promising solution over other low power NCL solutions (e.g. reference [12] and [13]), because of its possible compatibility in designing any combinational circuit.

In this paper, we exploited the NCL pipeline solution and port it into a conventional synchronous system. This work is based on the premise that if we develop standard cell libraries of NCL based low power soft error tolerant threshold gates, then they can become part of standard digital ASIC design flow. Our proposed solution use asynchronous (NCL based threshold gates) at micro level to implement combination logic, and maintain the synchronicity at macro level. Therefore, we named our technique as macro synchronous micro asynchronous (MSMA). We provided a complete timing analysis, and explained the details of implementation. Comparison with state-of-the-art soft error tolerant solutions

978-1-4799-3765-3/14 $31.00 © 2014 IEEE
DOI 10.1109/ISVLSI.2014.59

for purely synchronous systems are illustrated. It is observed that at 3 GHz clock frequency the proposed design performs almost 21% better, consumes at least 22% less power and has the 10% less energy delay product (EDP) as compared to the state-of-the-art soft error tolerant synchronous pipelines [14] [15].

The rest of the paper is organized as follows: Section II provides an overview of the proposed macro synchronous micro asynchronous (MSMA) pipeline architecture and its implementation. In Section III, we present the experimental results of this technique. In Section IV we discuss the analysis of the experimental results. Finally, Section V concludes the paper.

II. MACRO SYNCHRONOUS MICRO ASYNCHRNOUS (MSMA) PIPELINE

This section describes the overall timing perspective for a purely synchronous, asynchronous and hybrid (MSMA) pipeline. While describing the timing the architecture of the proposed design is also elaborated.

A. Synchronous Pipeline

Fig. 1 shows a conventional synchronous pipeline. It shows that there are two systems which are operating at frequency f. Both sender and receiver are sending, and receiving, the data at the positive clock edges. The total delay (T_{SYNCH}) for n number of pipeline stages can be written as equation (1). In the following equations T_{RISE} and T_{FALL} are the rise and fall time of the clock.

Suppose $T_{RISE} \approx T_{FALL}$

Therefore $T_{SYNCH} = nT$ (1)

If $(n-1)\{T - T_{RISE} - T_{FALL}\} < T_{CLB} + T_{R1} < n\{T - T_{RISE} - T_{FALL}\}$ (2)

Where, T_{CLB} and T_{R1} are the delays of combinational logic clock and register 1, respectively.

Contemporary soft error tolerant solutions require additional redundant hardware. In [14] triple modular redundancy (TMR) and C-Element based SE tolerant techniques are elaborated and using these techniques the area, power and delay of the system increases significantly.

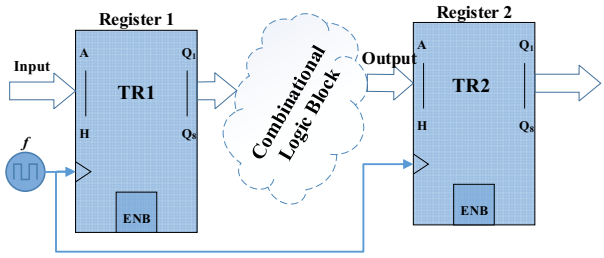

Figure 1. Single Stage Synchrnous Pipeline

Asynchrcnous Pipeline

Another method to implement the pipeline is to use purely asynchronous pipelines. NCL based asynchronous circuits have been proposed recently to tolerate soft error and also introduced low power techniques. In the following subsection we are providing a brief description and the timing of purely NCL based pipelines.

A traditional NCL pipeline architecture is primarily composed of three main blocks: NCL based registers, combinational logic block and completion detection scheme as shown in Figure 2. In this protocol, two stages interact through request and acknowledgement signals. In Figure 2, request and acknowledgement are based on Ki and Ko. This pipeline works in two different states: DATA state and NULL state. DATA state represents the DATA0 (0,1) and DATA1 (1,0) while NULL represents the NULL (0,0). If delay of register1, combinational block, register 2 and completion detection block are T_{R1}, T_{CLB}, T_{R2} and T_{CD}, respectively, then equation (3) provides the latency (T_{ASYNCH}), which is the duration from the availability of data to register 1 to the assertion of the acknowledgement signal through completion detection unit.

Therefore, $T_{ASYNCH} = T_{R1} + T_{CLB} + T_{R2} + T_{CD}$ (3)

NCL pipeline consists of two states of data communication: data and null states. These two states have different latencies, and in this paper we are naming them as data latency and null latency, respectively. It has been observed that transitions in data state of NCL makes the latency marginally higher than the null state, hence:

$$T_{DATA} \geq T_{NULL} \quad (4)$$

As the total latency of the pipeline should be sum of T_{DATA} and T_{NULL}. So the total Latency of this pipeline can be written as:

$$T_{TOTAL} = T_{DATA} + T_{NULL} \quad (5)$$

$$T_{TOTAL} \leq 2T_{DATA} \quad (6)$$

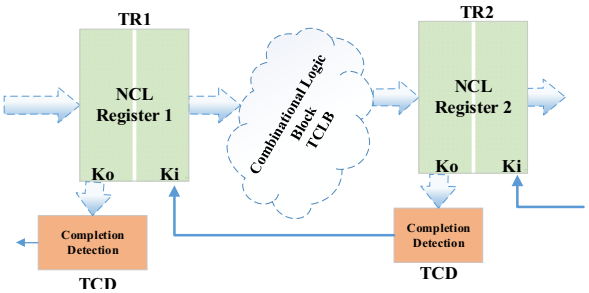

Figure 2. Single Stage Asynchrnous (NCL) Pipeline

B. Proposed Macro Synchronous Micro Asynchronous (MSMA) Pipeline

This is hybrid form of the synchronous and asynchronous pipelines. The following analysis assumes 4 stages of pipeline. In our proposed design, first and last stage of the pipeline is synchronous while all the other intermediate stages are asynchronous. So by implementing this pipeline we are able to make the asynchronous block (NCL Registers, combinational logic block and completion detection block) low power soft error tolerant by using the threshold gates of [11].In this paper

we are assuming that due to minimal area requirement of synchronous registers, completion detection, and dual rail to single rail (and vice versa) conversion are not soft error tolerant. For the sake of completeness, each of the above mentioned units can be either replaced by soft error tolerant synchronous registers and/or by limited version of TMR or C-element based soft error tolerant techniques.

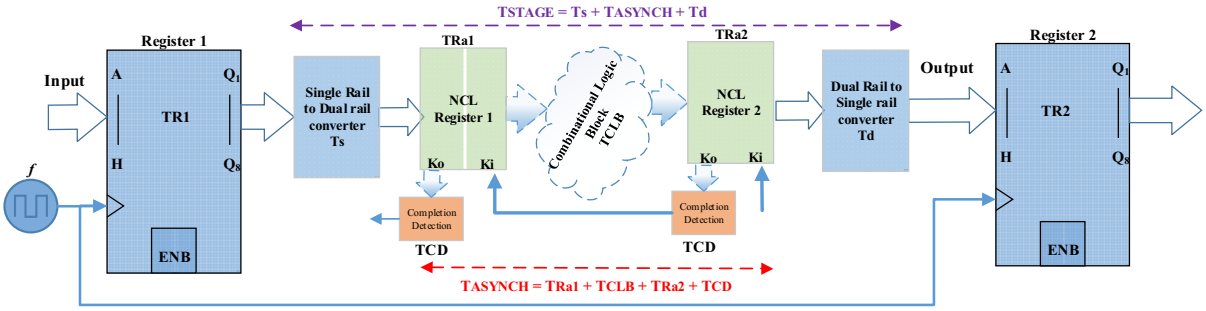

Figure 3. Macro Synchronous Micro Asynchronous Pipeline

The delay analysis of this pipeline is explained in the Figure 3 and using equation 7, 8. If the delay of asynchronous pipeline is $T_{ASYNCH} = T_{Ra1} + T_{CLB} + T_{Ra2} + T_{CD}$ where T_{Ra1}, T_{Ra2}, T_{CLB} and T_{CD} are delays of NCL registers, combinational logic block and completion detection block, respectively, then the total delay of MSMA (T_{MSMA}) is:

$$T_{MSMA} = nT \quad (7)$$

$$\text{If } (n-1)\{T - T_{RISE} - T_{FALL}\} < T_{STAGE} + T_{R1} < n\{T - T_{RISE} - T_{FALL}\} \quad (8)$$

Where $T_{STAGE} = T_S + T_{ASYNCH} + T_d$ (T_S and T_d are the delays of single to dual rail and dual to single rail conversion block, respectively) and "n" is the number of stages of pipeline.

III. Implementation Level Details of MSMA and Conventional Synchronous SE Tolerant Design

For the common ground comparison, we have also implemented the conventional synchronous pipelines with the SE tolerant solution. These are the typical synchronous pipeline, Triple Modular Redundancy (TMR) [14] based SE tolerant synchronous pipeline (TMRSS) and C-element based SE tolerant asynchronous (CESA) pipelines [15]. For all the designs 2 bit full adder is used as an example combinational logic block. The detailed implementation of our proposed design and other synchronous pipelines are given in the following sub sections.

A. Synchronous Pipelines

Fig. 1 shows that the synchronous pipeline consists of registers and combinational block. For implementation purposes, we used 2-bit adder as the combinational logic block.

B. TMRSS pipeline

In TMRSS pipeline, we used the standard method [14] of hardware redundancy. Hardware implementation of the TMR based design is given in Figure 4.

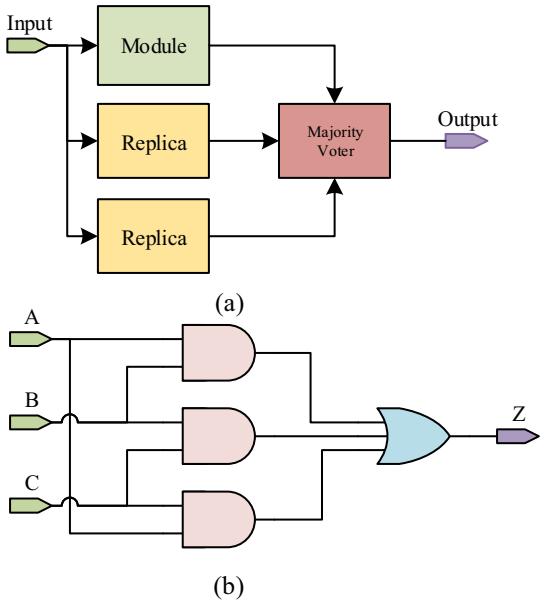

Figure 4. (a) TMR Implementation (b) Gate level implemenation of Majority Voter[14]

C. CESA pipeline

In CESA pipeline, the C-element is used to detect and avoid the glitches caused by the soft errors in the circuits. We used the cross coupled structure of C-elements for soft error tolerance [15] because it is more robust as compared to the typical methods [15]. Figure 6 shows the implementation of the Soft error tolerant method based on cross coupled structure of C-elements. The detailed hardware implementation can be found in [15].

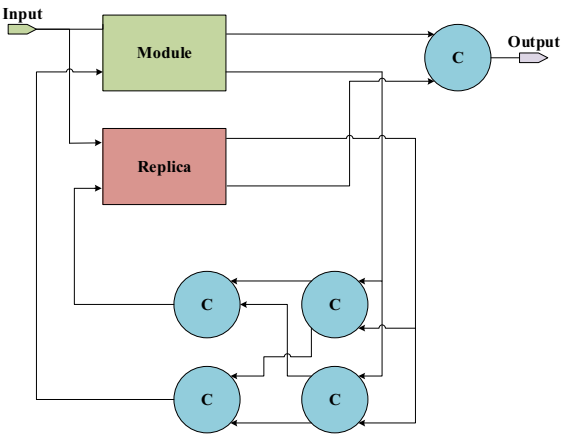

Figure 5. Cross coupled C-element based SE tolerant scheme[15]

D. MSMA Pipeline

The implementation of MSMA pipeline can be divided into two main blocks: Asynchronous and Synchronous. The asynchronous block includes the basic NCL pipeline which consists of three main blocks: Dual rail Registers, Combinational Block and Completion Detection. Since NCL is based on dual rail encoding therefore all these blocks are implemented using dual rail encoding. The synchronous block consists of synchronous registers at both ends of the systems shown in Fig. 3.

i. Asynchronous Block

This block includes the dual rail registers which are clock-less and thus the data receive and release operations are controlled by two signals Ko and Ki [9]. This register is composed of two basic threshold gates: TH22n and TH12b. It receives the data when Ki is "1" and sets the Ko when it receives the NULL. The combinational logic block comprises of 2 bit dual rail adder. The functionality of which can be explained from equation 9 & 10 [9].

$$\text{TH23:} \quad Y = A.B + A.C + B.C \tag{9}$$

$$\text{TH34w2:} \quad Y = A.B + A.C + A.D + B.C.D \tag{10}$$

As the dual-rail encoding considers the (0, 0) as a NULL state, therefore we must keep the carry in (C1 = 0, C0 = 1) to provide an equivalent of "0" in Boolean logic.

The Completion Detection unit generates the signal Ki which is used as ACK/NACK for the pipeline. Ki = 0 represents ACK and Ki = 1 represents NACK.

Two bit full adder is implemented in the combinational logic block. This adder is implemented by using the two low power soft error tolerant threshold gates TH34w2 and TH23, which was proposed by [11]. Figure 6 shows the optimized gate level implementation of the 1 bit NCL full adder [9]. 2 bit-adder is obtained using carry ripple adder, which is implemented using conventional arrangement of 2 full adders.

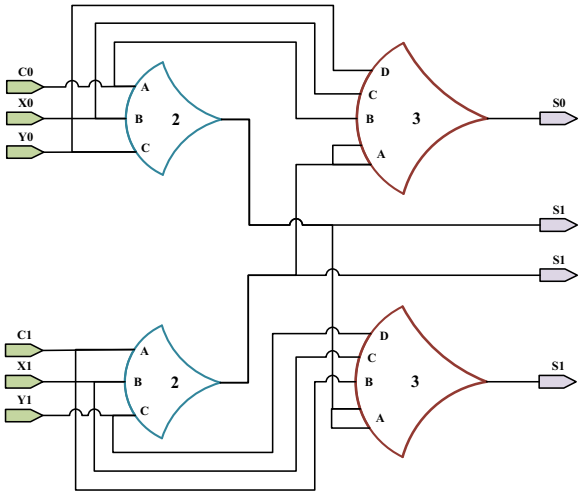

Figure 6. An Optimzed NCL Full Adder [9]

ii. Dual Rail to Single Rail (and Vice Versa) Conversion:

Single rail to dual rail converters convert the single bit data to dual rail. Similarly, dual to single rail converters convert the dual rail data into single bits. Gate level implementations are given in Figure 7 and 8. Figure 7 converts '10' to '1' and for all other signals it generates '0'. Figure 8 shows how to convert from single to dual rail. Because Z1 and Z0 need to match the timing, therefore transmission gate buffers are used.

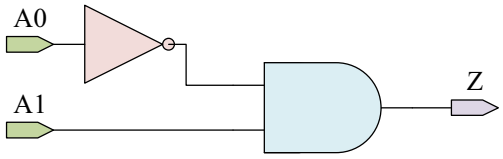

Figure 7. Dual to Single Rail converter

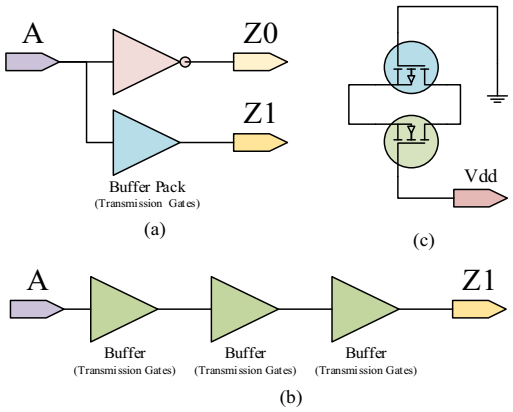

Figure 8. (a) Single Rail to Dual Rail Converter (b) Buffer Pack of Transmission Gates (to match the delays) (c) Transmission Gate Buffer

IV. Experimental Results

This section summarizes the results of the electrical simulations that are performed using 130nm CMOS technology. For the gate level implementation, semi static CMOS logic has been used. Transistor sizing in different

threshold gates, except the feedback inverter of each gate, has been done by using the conventional propagation delay matching of pull up and down networks. The feedback inverter is sized on the basis of gate capacitances of the feedback NMOS and PMOS. Latency and the average power consumption of the asynchronous system is analyzed. In order to find the robustness of the proposed pipeline method all the pipeline techniques discussed in Section III are implemented, simulated and analyzed at different frequencies. These analyses are summarized and discussed as follows.

A. Latency

i. MSMA Pipeline

As explained in the preceding section, that the NCL pipeline architecture works in two different states: NULL and DATA state. Therefore, the intermediate stage has the two different latencies: DATA latency and NULL latency, to complete DATA cycle and NULL cycle, respectively. This pipeline is enveloped by synchronous registers, which has traditional latency characteristics, and is described as the time difference between the time at which data is inserted to synchronous register 1 of Fig. 3 and the time at which synchronous register 2 of Fig. 3 receives the data. Fig. 9 shows simulation result of latency of MSMA pipeline with clock frequency of 3GHz. The time A in Fig. 9, refers to the time instant when the data is asserted in the system and B indicates the time instant when the data is received by register 2 of Fig. 3. Therefore, the time difference between the points A and B represents the latency of the system and is equal to 574ps. For this implementation, we also found out that in order to avoid skipping data changes, the data must remain valid for more than 574 ps at the input of sender.

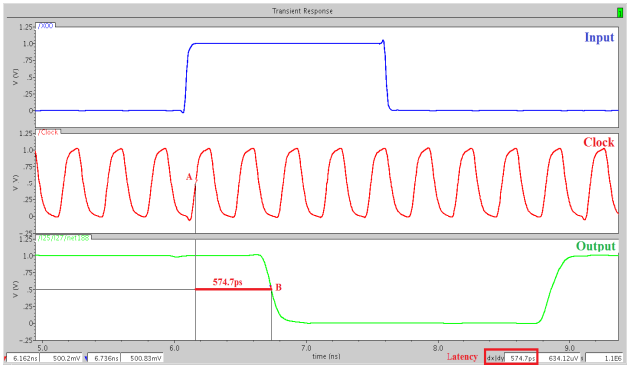

Figure 9. Latency of Macro Synchronous Micro Asynchronous Pipeline

ii. Other state of the Art Pipelines

As explained in the preceding section, the latency of the synchronous pipeline is dependent on the clock frequency. Therefore, its latency is defined as the time difference between the time at which clock samples to the time at which receiver receives the data. Figure 10, 11 and 12 shows the latency of the synchronous system without soft error tolerance, TMRSS and CESA pipelines, respectively. The frequency for all these cases is kept constant at 3GHz. Time A in the figure 8, refers the time instant when clock edge arrive and store the data at 1st register of Fig. 1. Similarly time B refers the time instant at which data is received by the last register of Figure 2. Therefore, the time difference between A and B is the latency of the synchronous pipeline. Similarly the Latencies of synchronous, MSMA, TMRSS and CESA are analyzed at different operating frequencies and these results are summarized in Table 1

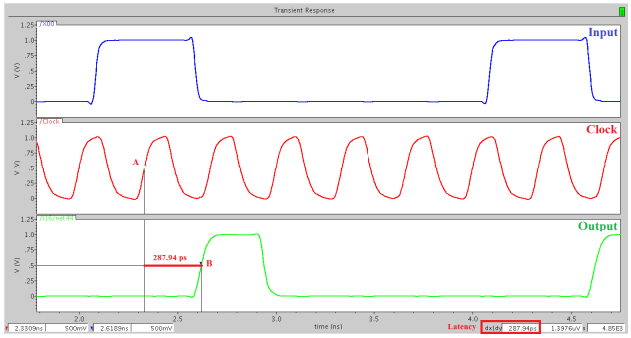

Figure 10. Latency of Synchronous Pipeline

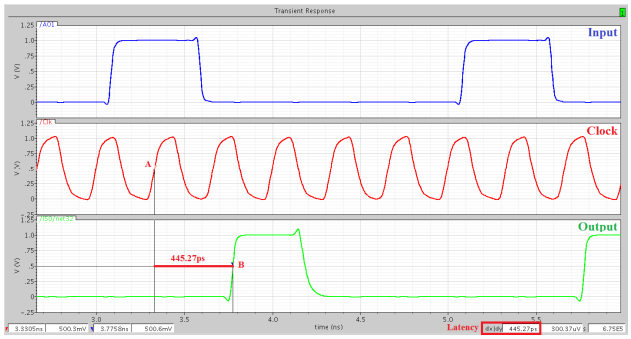

Figure 11. Latency of TMR based SE Tolerant Synchronous Pipeline

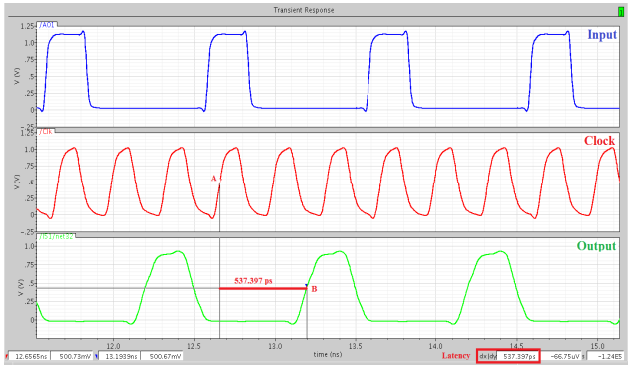

Figure 12. Latency of C-Element based SE Tolerant Synchronous Pipeline

TABLE I. LATENCY COMPARISON OF THE PIPELINES

Latency of the Pipelines				
***f* (GHz)**	***Synchronous Pipeline***	***TMRSS Pipeline***	***CESA Pipeline***	***MSMA Pipeline***
	Latency (D) (ps)	***Latency (D) (ps)***	***Latency (D) (ps)***	***Latency (D) (ps)***
1	*287*	*445*	*537*	*574*
2	*287*	*445*	*537*	*574*
3	*287*	*445*	*537*	*574*

B. Power

We utilized the worst case conditions (when data is transferring at every clock edge) for our power calculations and thus the values reported can be considered to be as the maximum power consumption values for the given circuit. The worst condition for this circuit occurs when the DATA hold time of the sender is equal to the DATA Latency. Table 2 shows the maximum power and average of the synchronous, TMRSS, CESA and MSMA pipelines.

TABLE II. POWER COMPARISON THE PIPELINES

Power Comparison of the Pipelines								
f (GHz)	*Synchronous Pipeline*		*TMRSS Pipeline*		*CESA Pipeline*		*MSMA Pipeline*	
	Max. (μW)	*Avg. (μW)*	*Max. (μW)*	*Avg. (μW)*	*Max. (μW)*	*Avg. (μW)*	*Max. (μW)*	*Avg. (μW)*
1	*620*	*134*	*759*	*200*	*727*	*174*	*600*	*135*
2	*724*	*216*	*810*	*260*	*790*	*251*	*693*	*201*
3	*867*	*291*	*957*	*336*	*941*	*324*	*853*	*279*

V. DISCUSSION

Table 1 shows that the latency of our proposed MSMA pipeline is 23% more than the TMR based synchronous pipelines. This additional latency is induced due to the addition of converting blocks. However, if multiple synchronous stage pipelines are replaced by NCL pipelines then this latency may become insignificant. This difference in the latencies can also be reduced by optimizing the all operational blocks.

Table II shows that the power consumption of the pipeline is significantly lower than the soft error tolerant synchronous pipelines (TMRSS and CESA). Power analysis in Table 2 shows that MSMA pipeline is consuming 22% and 32% less power than CESA and TMRSS pipelines, respectively.

Table III shows Energy Delay Product (EDP) comparison, which indicates that the MSMA pipeline is 10% and 21% better than the other soft error tolerant synchronous pipelines (TMRSS and CESA pipeline) respectively. This analysis also shows that at 3 GHz MSMA consumes almost 31% and 23% less energy as compared to TMRSS and CESA, respectively

TABLE III. ENERGY DELAY PRODUCT COMPARISON THE PIPELINES

Energy Delay Product Comparison of the Pipelines								
f (GHz)	*Synchronous Pipeline*		*TMRSS Pipeline*		*CESA Pipeline*		*MSMA Pipeline*	
	E (nJ)	*EDP* 10^{-18}	*E (nJ)*	*EDP* 10^{-18}	*E (nJ)*	*EDP* 10^{-18}	*E (nJ)*	*EDP* 10^{-18}
1	*17*	*5*	*40*	*17*	*30*	*16*	*18*	*10*
2	*46*	*13*	*67*	*30*	*63*	*33*	*40*	*23*
3	*84*	*24*	*112*	*49*	*105*	*56*	*77*	*44*

VI. CONCLUSIONS AND FUTURE WORK

Soft errors are a severe threat to the reliable functionality of the modern digital system. In an attempt to design a low-power and soft error tolerant digital systems, this paper presents a pipeline which is broadly synchronous and locally asynchronous and called as macro synchronous micro asynchronous (MSMA). A complete framework of the methodology and its timing analysis is provided. Low-power is attained in MSMA with the use of soft error tolerant threshold gates based on NCL logic. Electrical simulation results indicate an improvement of at least 10 % in EDP metric over conventional soft error tolerant solutions for synchronous designs. Power analysis indicates that our proposed MSMA uses at least 22% less power compared to [14] and [15]. In future, we intend to explore on how to reduce the latency and make such pipelines as part of standard cell based digital design flow.

REFERENCES

[1] Mukherjee, Shubu. Architecture design for soft errors. Morgan Kaufmann, 2011. ISBN: 978-0-12-369529-1.

[2] P. Shivakumar, M. Kistler, S. W. Keckler, D. Burger and L. Alyisi, "Modeling the Effect of Technology Trends on the Soft Error Rate of Combinational Logic," in the Proceedings of the IDS, 2002 pp. 1-6.

[3] Y. Monnet, M. Renaudin and R. Leveugle, "Asynchronous circuits transient faults sensitivity evaluation", in DAC, Anaheim, CA, 2005, pp. 863–868.

[4] Y. Monnet, M. Renaudin and R. Leveugle, "Asynchronous circuits sensitivity to fault injection," in Proc. 10th IEEE Int. On-Line Testing Symp., 2004, pp. 121–126.

[5] Y. Monnet, M. Renaudin and R. Leveugle, "Hardening Techniques against Transient Faults for Asynchronous Circuits," in Proc. 11th IEEE Int. On-Line Test. Symp., 2005, pp. 129–134.

[6] W. Jang and A. J. Martin, "SEU-Tolerant QDI circuits," in Proc. IEEE Int. Symp. Asynchronous Circuits Syst., 2005, pp. 156–165.

[7] S. Peng and R. Manohar, "Efficient Failure Detection in Pipelined Asynchronous circuits," in Proc. IEEE Int. Symp. Defect Fault Tolerance VLSI Syst., 2005, pp. 484–493.

[8] K.T. Gardiner, A. Yakovlev and A. Bystrov, "A C-element Latch Scheme with Increased Transient Fault Tolerance for Asynchronous Circuits", IOLTS 2007, pp.223-230

[9] W. Kuang, P. Zhao, J. S. Yuan, and R. F. DeMara, "Design of Asynchronous Circuits for High Soft Error Tolerance in Deep Submicrometer CMOS Circuits," IEEE Trans. On VLSI, vol. 18. 2010, pp. 410-422.

[10] F. K. Lodhi, O. Hasan, S. R. Hasan and F. Awwad "Modified Null Convention Logic Pipeline to Detect Soft Errors in Both Null and Data Phase " in IEEE International Midwest Symposium on Circuits and Systems (MWCAS 2012), pp 402 - 405.

[11] M. Mosaffan, F. Jafari, S. Mohammadi, "Designing Robust Threshold Gates against Soft Errors," Microelectronics Journal, vol 42, Issue 11, 2011, pp. 1276-1289.

[12] A. Bailey, A. Zahrani, G. Fu, J. Di and S. Smith., " Multi-Threshold Asynchronous Circuit Design for Ultra-Low Power," Low Power Electronics, vol. 4/3, pp. 337-34S, 200

[13] L. Zhou, S. Smith, and J. Di, "Bit-Wise MTNCL: an Ultra-Low Power Bit-Wise Pipelined Asynchronous Circuit Design Methodology," IEEE Midwest Symposium on Circuits and Systems, August 2010

[14] E. Dubrova, "Hardware Redundency", In Fault Tolerant Design, Springer-Verlag New York Inc., New York, NY, 2013

[15] S. Almukhaizim, S. Feng and Y. Makris, "Coping with Soft Errors in Asynchronous Burst-Mode Machines" IEEE International Symposium on Asynchronous Circuits and Systems, ASYNC, pp. 151-160, April 2008.