

Functional Enhancements of TMR for Power Efficient and Error Resilient ASIC Designs

Hagen Sämrow, Claas Cornelius, Philipp Gorski, Jakob Salzmänn, Andreas Tockhorn and Dirk Timmermann

Department of Electrical Engineering, University of Rostock, Rostock, Germany

Email: hagen.saemrow@uni-rostock.de

Abstract—Progressive technology scaling raises the need for efficient VLSI design methods facing the increasing vulnerability to permanent physical defects, while considering power efficiency of resulting circuit implementations at the same time. Triple Modular Redundancy (TMR) represents a common method to encounter reliability problems, but has the drawback of increased area and power consumption. This work introduces a Low Power Redundant (LPR) design solution that targets the power penalty of TMR implementations. This is done by enhanced and new functional runtime capabilities for error detection and operation control. By exploiting the inherent modularity and parallelism of TMR, the LPR solution applies additional control logic to switch dynamically between compare phases (to indicate faults and their locations) and parallel operation (with reduced operation frequency). This allows power optimized circuit operation with full support for the treatment of permanent faults. Simulation results on different ALU implementations show a decrease of power consumption of up to 60 % compared to conventional TMR. Furthermore, different strategies for the switching between operation modes are introduced that enable power efficient system operation in the presence of permanent physical defects. Moreover, significant reliability improvements are also achieved due to the adaptive use of the redundant modules.

Keywords: *Circuit Design, Reliability, Triple Modular Redundancy, Power Consumption, Power-Aware Design*

I. INTRODUCTION

The ongoing technology scaling in the semiconductor fabrication process results in reliability and power issues that are threatening for the design of integrated circuits. Various reliability issues have worsened to unacceptable levels due to rising power and current densities which increase temperature levels and gradients. Hence, this trend further accelerates the negative impact on long term wear-out effects [1] [2] that are introduced briefly in the following.

Tunneling currents into transistor gate oxides result in short connections through these transistor gates after a certain time of operation. High gate voltages and temperatures increase the degree of these defects. This long term effect is called Time-Dependent Dielectric Breakdown (TDDB) which leads rather to delay failures than to immediate malfunctions [3]. However, erroneous outputs will be produced in the presence of diverse defective transistors due to the disarrangement of the circuit's timing behavior [4]. Further gate oxide defects resulting in delay failures and finally in permanent functional failures are caused by trapped electric carriers (hot carriers) [5] and

Negative Bias Temperature Instability (NBTI) that increases the threshold voltage of p-MOSFETs [6].

Moreover, electromigration is the transport of wire material alongside the electric current flow, which results in short connections and increased wire resistances or even open wires [7]. All aforementioned effects deteriorate with rising temperature. Other stress effects like self-heating [8] and thermal cycling [9] also initiate malfunctions of integrated circuits due to fast temperature changes. Therefore, tool assisted reliability improvements will be a key priority in future VLSI designs to establish common solutions at the design stage for these comprehensive problems [10].

A very general and universal method for the treatment of faults is Triple Modular Redundancy (TMR). In this case, a circuit module is triplicated and the corresponding output vectors are merged through a majority voter [11] [12]. Several investigations aimed at enhancing this basic idea. For example, an increase of data integrity is achieved in [13] by changing the voting scheme and a totally self checking TMR circuit is presented in [14]. The main disadvantages of TMR are roughly threefold area and power consumption. Especially, the increased power consumption leads to higher heat dissipation with potentially severely damaging impacts [15] because lifetime of integrated circuits decreases exponentially with temperature, as simply illustrated by the Arrhenius relationship [16].

One general method to face increased power consumption is the usage of parallel data paths [17]. Our contribution proposes an architecture which uses the existing parallel modules of TMR in order to exploit the power savings of parallel data paths. Thereby, the modules operate in two alternating modes: a phase of parallel operation with reduced frequency and a compare phase to detect failures. Further on, an additional TMR mode allows the localization of faulty modules. The reduction of the number of active modules and of the operating frequency significantly decreases the power consumption compared to conventional TMR without the need for dynamic voltage regulation. Focusing on wear-out mechanisms due to the increasing threat of permanent defects for integrated circuits, soft error prevention is out of scope for this contribution. However, the presented solutions are able to detect permanent faults and to mask and to retard the aforementioned wear-out effects. Due to a fixed schedule of alternating compare phase and parallel operation, our solution provides an opportunity for the hardware to recover to an errorless state after a certain time. Unlike other approaches which minimize energy consumption by early optimizing TMR

and duplex systems based on high level modeling [18] [19], this work tackles the problem at the gate level. This also allows a seamless integration of the proposed design modification into an automated EDA workflow. In [20] we presented a related approach where the design has been toggled between a TMR and a standalone mode. We have reached power savings up to 50 %, but we have not delved into flexible strategies when faults are detected. Such promising results serve as the starting point for the in-depth investigation here.

The remainder of this paper is organized as follows: Section II highlights the concept of this contribution and the implementation of the enhanced designs. Section III presents the synthesis results for a 65 nm process technology. The final conclusions and an outlook will be given in Section IV.

II. DESIGN IMPLEMENTATION AND OPERATION MODES

A. Design Idea and Operation Modes

In order to reduce the power consumption of conventional TMR, functional enhancements are inserted to enable the proposed LPR designs to operate in different modes of operation. Such modes which are depicted in Figure 1 depend on the detected fault conditions:

If no fault is present: The design runs in the Dual Mode (DM), whereas only two of the three redundant modules are active. This reduces the power consumption and preserves the remaining idle unit. In addition, those two active modules toggle alternately with a configurable fixed time interval between two types of operation. More precisely, both active modules run for a given time in parallel operation and in the compare phase. Running in parallel operation means that each of the two active modules calculates independently different input data so that the number of finished calculations per cycle is doubled. The other way round, this allows to cut the frequency in half without a loss in calculation performance, while lowering the power consumption in comparison to conventional TMR. By contrast, the compare phase assigns the same input data to both operating modules which operate with the original frequency. After the computation, the outputs are compared to detect faults within the modules. It should be noted that the third idle module is changed after each compare phase. Higher stages of the design hierarchy have to be aware

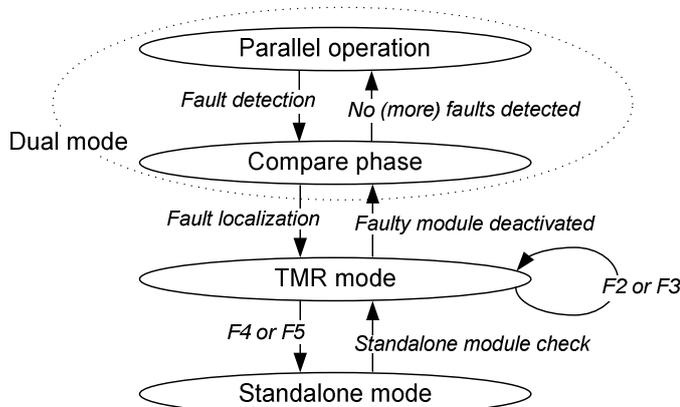


Figure 1. Implemented modes and possible changeovers between these modes

of possible problems which could occur because faults could not be detected immediately.

If one fault is present: The design has detected an erroneous situation during the Dual Mode (DM). Thus, the fault has to be localized and identified for dedicated reactions of the control logic. For that case the module changes to TMR mode, where all three modules run in the compare phase. Hence, the outputs are voted and compared in the Error Detection Unit. If a faulty module is localized in the TMR mode, the remaining two correctly working modules return to the DM and the faulty module is set to idle.

If more faults are present: If the control logic indicates another fault while the remaining two modules are running in DM, five different finalization strategies ($F1 - F5$) are possible:

$F1$: The design completely shuts down all operations.

$F2$: The design operates continuously in TMR mode. This strategy assumes that the faults in each module affect different output bits. Thus, the majority voter can correct each triplet of output bits in the TMR mode and the design proceeds with all three modules.

$F3$: This strategy works as $F2$, but the entire design is shut down when at least one fault is identified in each of the three modules.

$F4$: A fourth strategy idles the two defect modules. The remaining module runs standalone with the original frequency.

$F5$: Lastly, $F4$ is extended such that the entire design is shut down when an error is detected in the remaining standalone module during inserted TMR mode phases.

The proposed finalization strategies are evaluated regarding their drawbacks and advantages in section III C.

B. Design and Control Structures

Five different implementations of Arithmetic Logic Units (ALU) have been chosen as reference designs (see TABLE I.). This allows the evaluation of all implementations regarding different circuit sizes and performance penalties. These reference designs (ALU1 to ALU5) have been used to implement conventional TMR solutions and the proposed LPR designs (LPR1 to LPR5, for example LPR of ALU1 = LPR1). In addition to the TMR designs, the LPR designs are expanded with the new control units to switch between the different operation modes. Furthermore, error detection structures were added to localize a defective module or faults in the control units. Figure 2 illustrates the design of the redundant modules and the additional circuitry.

TABLE I. SYNTHESIS RESULTS OF REFERENCE DESIGNS (ALU1 TO ALU5) FOR A 65 NM TECHNOLOGY PROCESS LIBRARY

Design characteristics	Designs				
	ALU1	ALU2	ALU3	ALU4	ALU5
# of output bits	32	64	64	96	96
# of cells	2335	7078	11839	15542	23093
Minimum delay [ns]	2.63	8.36	4.66	12.51	6.72

The Demultiplexer Unit assigns input data to the corresponding modules. This is not implemented by individual demultiplexers for each bit, but by setting the appropriate enable signals of the input registers of the affected modules. For this reason, changes at the original clock domain are circumvented because the design clock remains the same for all designs (ALU_x, TMR_x, LPR_x).

The outputs of the triplicated modules have to be multiplexed according to the operation mode, which is managed within the Multiplexer Unit. For example, the outputs of the modules are voted if the design operates in TMR mode or the appropriate output of a module is directly forwarded to the design output if it runs in other modes.

During the parallel operation of the Dual Mode (DM) the Demultiplexer Unit enables the input registers of the two involved modules alternately in such a way that each module receives new input data with every second clock cycle. Accordingly, the results of the two modules are also alternately multiplexed to the design's output. Hence, one data set is still available every clock cycle at the design output. During the compare phase of DM, the involved modules access the same input data and one of the two results is forwarded to the design output.

It should be noted that the latency of the LPR designs increases from one to two clock cycles. This is required to compensate for the operation mode changes.

C. Error Detection

Designing the Error Detection unit, different states of detection have to be considered. First, faults in the modules should be indicated during the compare phase of DM. Second, the locations of the faults have to be identified during TMR mode. Figure 3 depicts an example of the error detection in the LPR designs for one bit (here module M3 is idle in Figure 3a). If two modules are operating in the compare phase (Figure 3a) an XOR-stage determines bit by bit if differences exist between the modules. If so, a fault signal is directed into an OR-tree which is implemented threefold (for every module, one tree is available). These OR-trees merge all possible bit faults of one module to one signal indicating a fault for each module individually. The interposed AND-stage ensures that only fault signals of operating modules are forwarded to the OR-trees. Signals of idle modules will be set to a logic 0.

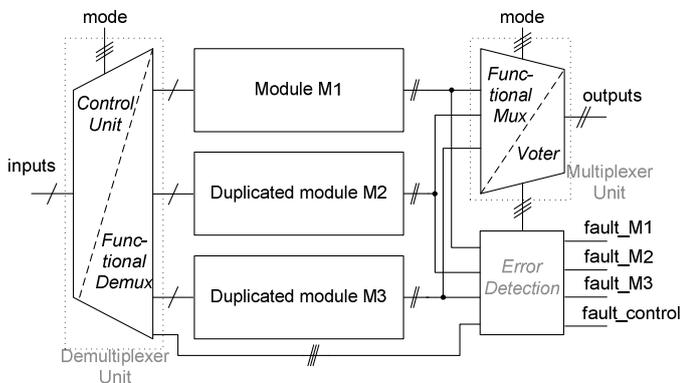


Figure 2. LPR design schematic with additional control logic: De-/Multiplexer and Error Detection Unit

During TMR mode (Figure 3b) a second AND-stage is involved to identify the defective module. The first AND-stage is transparent because all enabling signals are on. The second AND-stage merges the fault signal combinations (M1/M2, M1/M3, M2/M3) of the XOR-stage to one fault signal which identifies the defective module (M1, M2 or M3). This is achieved due to the conjunction of the different fault signal combinations, which is depicted in TABLE II.

TABLE II. MAPPING OF THE CORRECT DEFECTIVE MODULE

Fault signal combination 1		Fault signal combination 2	Indicating Defective Module
M1 xor M2	AND	M1 xor M3	M1
M1 xor M2	AND	M2 xor M3	M2
M1 xor M3	AND	M2 xor M3	M3

Further mechanisms have been implemented to observe the correct behavior of control signals themselves. Especially stuck-at-0 faults in the OR-tree would be disastrous because detected defects could be masked. Hence, if faults are detected in the control units, the whole design is shut down immediately. On the one hand, during the compare phase of DM, the two OR-trees of the active modules are connected to the same fault signals from the first AND-stage. Thus, by comparing the outputs of both OR-trees, a difference indicates a defective OR-tree. By doing so, the stuck-at-0 faults can be detected with this preventive measure. On the other hand, the enable signals for the input registers and the output assignments are implemented twice (one in each control unit). Differences in the enabling signals indicate a fault in one of the two units.

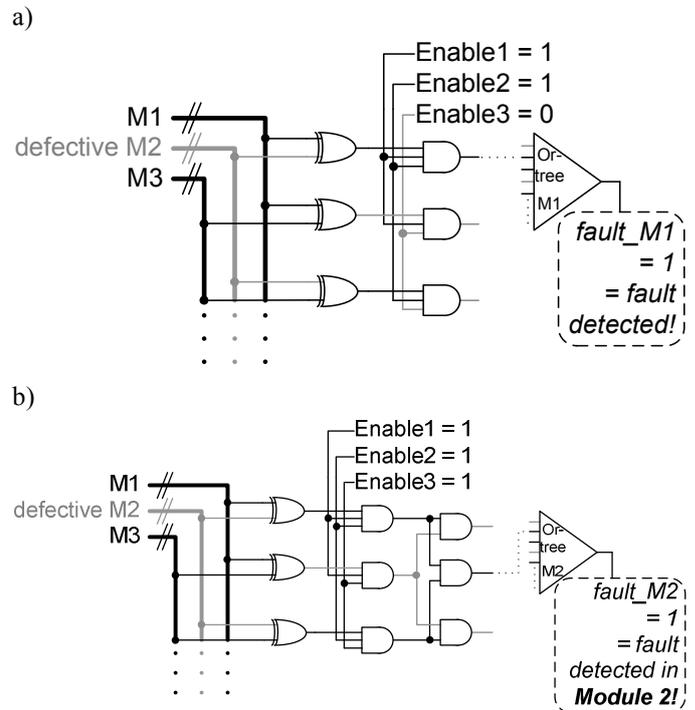


Figure 3. Example to illustrate the working principle of the Error detection unit depicted for one bit (signal = 1 → black line, signal = 0 → grey line):

- a) During the compare phase (here, M1 und M2 are on)
- b) During TMR mode (all modules are on)

III. EXPERIMENTAL RESULTS AND DISCUSSION

A. Simulation Setup

The proposed designs of section II have been implemented and synthesized using an industrial 65 nm standard cell ASIC library. The resulting netlists have been used to apply, verify and evaluate the enhancements at gate level. Hereby, the analysis includes area and delay, but focuses on reliability and power consumption. For reliability analyses, stuck-at faults have been induced into the netlists during the functional simulations to model permanent wear-out effects. The locations of the defects have been randomly chosen and are based on a constant failure rate λ for every net of the designs. This modeling strategy results in a realistic and area dependent failure rate for the whole design. More precisely, small designs are less often affected by failures over a given period of time than large designs. Furthermore, the power values of the whole designs (including all additional control) have been obtained by the average of multiple simulation runs considering the individual toggle rates of all nets.

B. Simulation Results without Fault Injection

Figure 4 contains the relative overhead in chip area and resulting delay of the LPR designs compared to the synthesis results of conventional TMR. It shows that the proposed LPR solution results in higher area costs and delay penalties. The integration of additional logic for control and error detection increases the number of design cells depending on the width of the output vector and influences the critical path with a fixed additional delay. Due to the fixed delay, the relative penalty is higher for faster designs (LPR1, LPR3), while slower designs (LPR2, LPR4) suffer less. Moreover, because the voting and error detection structures operate bit by bit, more output bits result in more area overhead. However, this relationship has to be compared with the overall area of the designs. As it can be seen in the figure the relative area overhead is smaller for the larger designs (LPR3 and LPR5). By contrast, the smaller designs (LPR1 and LPR2) exhibit the largest additional area overhead.

However, these penalties come with the benefit of drastically decreased power consumption compared to

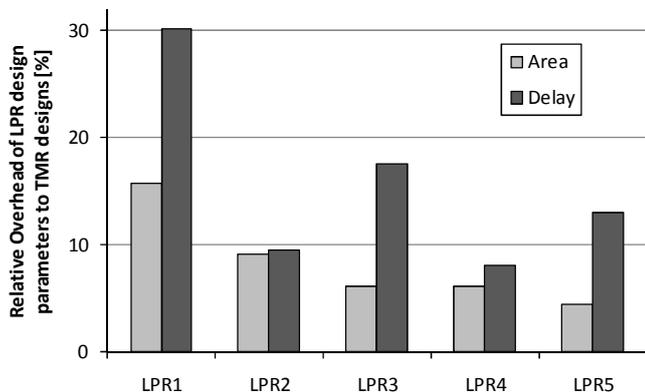


Figure 4. Design overhead of the LPR solution in relation to a conventional TMR implementation

conventional TMR design. Figure 5 illustrates that the LPR designs decrease the power consumption down to 40 % in the best case. In this case, the power savings are clearly higher than the increase of implementation costs. Thus, it represents a power optimized tradeoff of design parameters.

More precisely, Figure 5 plots the power consumption of the different designs over the relative time that the design operates in the compare phase p_{COMP} of the DM. As described in section II, the LPR designs in DM alternately toggle between parallel operation and the compare phase as long as no fault occurs. For instance, $p_{COMP} = 0$ means that the modules run exclusively in parallel, whereas $p_{COMP} = 1$ denotes that the modules operate solely in the compare phase. The duration of the compare phase has been set to an appropriate level so that the design is able to detect the faults in most cases [20]. In Figure 5 all power values are calculated in relation to the conventional TMR design. However, an additional design is included in the figure, which represents a single module without any redundancy. The power consumption of this reference design (REF) is independent of p_{COMP} and thus constant over the plotted interval.

The power savings related to TMR depend linearly on p_{COMP} with the largest savings for the smallest possible parameter p_{COMP} , because at this point the module works most of the time in parallel operation with halved operation frequency. Here, large designs benefit the most from the LPR enhancements. Strictly speaking, the largest analyzed design (LPR5) requires only 40 % of power consumption compared to conventional TMR for $p_{COMP} = 0.02$. This is close to the reference design with only one module, which consumes on average 33 %. However, even for the worst case of $p_{COMP} = 1.0$, the LPR solutions still require only 75 % of power in relation to conventional TMR, since one of three TMR modules is idled.

Summarizing, the simulation runs without fault injection clearly show that the LPR designs outperform conventional TMR. Further on, larger designs with applied LPR exhibit greater power savings. But even the smallest design (LPR1) is able to decrease power consumption of up to 50 %.

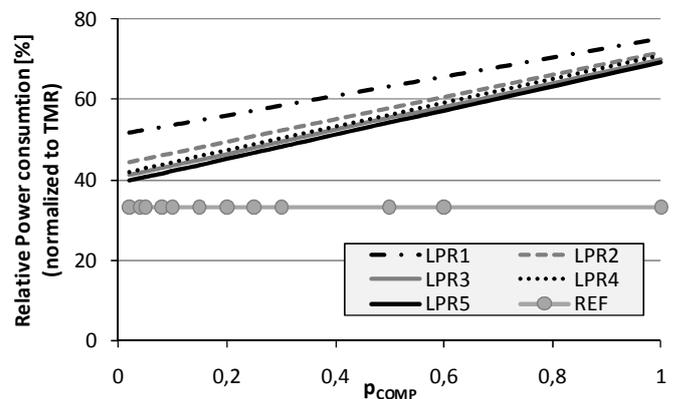


Figure 5. Power consumption in the absence of faults of the reference and LPR designs plotted over the p_{comb} parameter (in relation to the conventional TMR = 100 %)

C. Simulation Results with Fault Injection

The increased area needs for the LPR implementations also have a meaningful impact on reliability. Therefore, manifold reliability simulations for the LPR designs including all five possible finalization strategies were executed with different constant failure rates. In this perspective, the reliability $R(t)$ of a design is understood as the probability of a design to perform as desired until time instance t . To evaluate the reliability $R(t)$ of a design here, the time stamps were recorded when erroneous design outputs occurred or when a shutdown was initiated by the control logic of the LPR designs. Consider that the LPR designs are only able to detect errors during the compare phase of DM or in TMR mode. Thus, the following assumptions have been made:

- If the system is operating in the compare phase of DM and has not reacted appropriately to an erroneous output during the previous parallel operation, the time stamp for the first faulty output is recorded for the reliability measurement.
- If the control units of the LPR designs are able to detect the fault and no finalization strategy was previously active (i.e. a maximum of one fault occurred), the design continues its operations until the first error that is not recognized.

Figure 6 depicts the simulated reliability measurements exemplary for the synthesized ALU3 designs (for $p_{\text{COMP}} = 0.1$), since these results are also representative for the simulation runs of the other designs. The figure represents the average observed Mean Time To Failure (MTTF) of different simulation runs across varying failure rates and different finalization strategies. The MTTF represents the average time a system operates until it fails. It is equal to the expected lifetime if the system cannot be repaired, whereas the MTTF can be calculated by [21]:

$$MTTF = \int_0^{\infty} R(t) dt \quad (1)$$

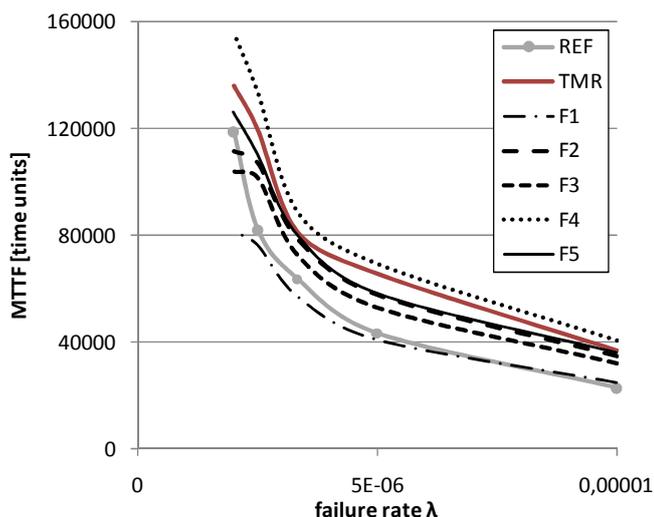


Figure 6. MTTF of the ALU3 designs (REF, TMR, LPR) dependent on the failure rate

The simulations with fault injection show that for low failure rates the MTTFs of all simulated designs are quite different. In fact, the necessary overhead in contrast to the reference design does only pay off for the conventional TMR and the finalization strategies $F4$ and $F5$. However, for most simulated scenarios all designs outperform the reference design in terms of reliability. More precisely, the controlled shutdown after detecting two defect modules (in case of $F1$) results in slightly lower reliabilities than the reference design across the entire range of failure rates. Nonetheless, the advantage still remains concerning the reduced power consumption (see section before).

Furthermore, the two strategies $F2$ and $F3$ which finally run in TMR mode show a slightly decreased MTTF compared to the conventional TMR but a better MTTF than the reference design. The MTTF of $F3$ is 5 % to 10 % smaller than the one of $F2$. This results from the controlled shutdowns in $F3$, because faults in all three modules do not necessarily lead to an erroneous design output when different bit positions are affected by the faults. From a reliability perspective, a design benefits most when it finally switches to the last correctly working module ($F4$ or $F5$). This leads to the greatest achievable MTTF values. Furthermore, due to a possible incorrect detection of faults at the same output bits caused by the two other defective modules, the MTTF of $F4$ is higher than the MTTF of $F5$ because the final strategy $F5$ shuts the system down in the presence of three defective modules.

The simulated LPR designs with $F4$ as the finalization strategy achieved a 5 % to 15 % higher MTTF than conventional TMR systems. This is attributable to the smaller fraction of chip area that is affected by wear-out defects in case of $F4$ in contrast to the three modules of TMR. In conclusion, for higher failure rates almost all LPR designs (except $F1$) increase the reliability in comparison to the reference design, whereas the finalization strategy $F4$ even outperforms the conventional TMR design.

Figure 7 exemplarily illustrates the power consumption over time when faults occur for an ALU2 design and all implementations. The illustrated changes of the operation

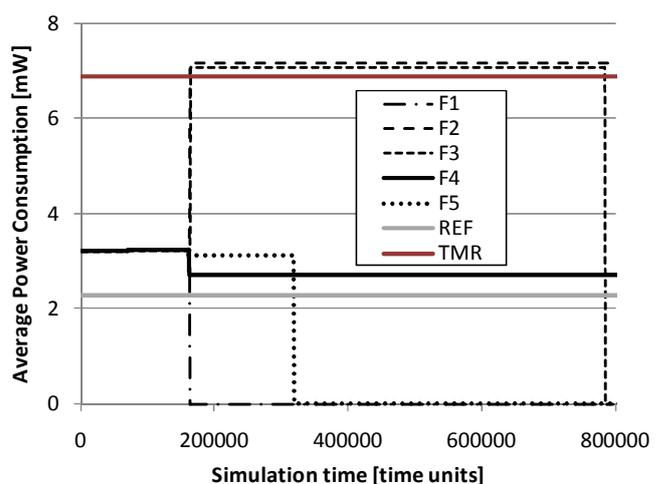


Figure 7. Power consumption of a average ALU2 design ($\lambda = 1/500000$, $p_{\text{COMP}} = 0.1$)

modes (i.e. equal to changes in power consumption here) are based on average values from the reliability simulation runs. It can be seen that the power consumptions of the different finalization strategies (*F1-F5*) are identical before $t = 162000$ time units. From this time on, two of the three modules are already identified as erroneous and the finalization strategies are executed. Hence, the advantages of *F1*, *F3* and *F5*, regarding the power consumption, appear very clearly. While *F1* shuts the system down immediately, *F3* and *F5* continue until the last working module becomes faulty. Simply spoken, all three strategies stop to operate (and thus to dissipate power) after a certain amount of faults has occurred which is also used to indicate the outage to the superior system. *F2* and *F3* consume slightly more power in the presence of faults than conventional TMR caused by the additional control overhead. The same conditions result in slightly higher power consumption of *F4* and *F5* compared to the reference design (REF).

Summarizing, the failure simulations show that the benefits of LPR designs strongly depend on the selected finalization strategy. The strategies that also operate with only one remaining working module (*F4*, *F5*) are more power efficient and achieve larger MTTF values than the approaches with a final TMR operation mode (*F2*, *F3*). Thus, the finalization strategy *F5* is the best suited, especially for mobile applications due to the shutdown ability in contrast to *F4*. Strategy *F1* appears solely useful for a very reliable system in order to reduce power consumption after faults have occurred.

IV. CONCLUSION AND OUTLOOK

This work proposes a convenient approach to the growing challenges of power efficient and error resilient circuit designs for shrinking technologies. Therefore, the conventional TMR approach is enhanced with additional control logic and error detection. This allows a runtime adaptive operation as regards permanent defects as well as dedicated features for the control of power consumption. During the parallel operation of the Dual Mode (DM) the operation frequency of the modules is halved, enabling the system to consume less power than conventional TMR. Moreover, permanent defects and their locations can be detected during the available TMR mode. The drawback is an increased area overhead, a slight delay penalty and the loss of transient fault prevention. However, power savings of up to 60 % compared to conventional TMR can be achieved while permanent errors are detectable.

Future investigations will concentrate on additional power reductions based on dynamic voltage scaling during the parallel operation of DM. Furthermore, the automated integration into existing EDA design flows at gate level will be considered as well.

REFERENCES

- [1] Mc Pherson, J.W.: "Reliability trends with advanced CMOS scaling and the implications for design", In Proc. of Custom Integrated Circuits Conference, San Jose, USA, 2007.
- [2] G Srinivasan, J., Adve, S., Bose, P. and Rivers, J., "The Impact of Technology Scaling on Lifetime Reliability", In Proc. of Dependable Systems and Networks, Florence, Italy, 2004.
- [3] Vogel, E.M., Suehle, J.S., Edelstein, M.D., Wang, B., Chen, Y.: "Reliability of Ultra-Thin Silicon Dioxide Under Combined Substrate Hot Electron and Constant Voltage Tunneling Stress", In IEEE Trans. On Electron Devices, 2000.
- [4] Kaczer, B., Degraeve, R., Roussel, P., Groeseneken: "Gate Oxide Breakdown in FET devices and circuits: From nanoscale physics to system-level reliability", In Microelectronics Reliability, Elsevier, 2007.
- [5] Li, P.-C., Stamoulis, G., Hajj, I.N.: "A probabilistic timing approach to hot-carrier effect estimation", In Trans. On International Conference on Computer-Aided Design, Santa Clara, USA, 1992.
- [6] Schröder, D.; Babcock, J.: "Negative Bias Temperature Instability: Road to Cross in Deep Submicron Silicon Semiconductor Manufacturing", In Journal of Applied Physics, vol. 94, 2003.
- [7] Srinivasan, J.; Adve, S.; Bose, P.; Rivers, J.: "The Case for Lifetime Reliability-Aware Microprocessors", In Proc. of 31st International Symposium on Computer Architecture (ISCA), Munich, Germany, 2004.
- [8] Tenbroek, B.M., Lee, M.S.L., Redman-White, W., Bunyan, R.J.T., Uren, M.J.: "Self-heating effects in SOI MOSFETs and their measurement by small signal conductance techniques", In IEEE Transactions on Electron Devices, 1996.
- [9] Gardner, D.S., Flinn, P.A.: "Mechanical stress as a function of temperature in aluminum films", In IEEE Transactions on Electron Devices, 1988.
- [10] Semiconductor Industry Association (SIA), "International Technology Roadmap for Semiconductors", Release 2009.
- [11] Von Neumann, J.: "Probabilistic logics and the synthesis of reliable organisms from unreliable components", In Automata Studies, Princeton University Press, 1956.
- [12] Siewiorek, D.: "Architecture of fault-tolerant computers: An historical perspective", In Proc. of the IEEE 79 (12), 1991
- [13] Mitra, S., Mc Cluskey, E.J.: "Word-voter: a new voter design for triple modular redundant systems", In Proc. of the IEEE VLSI Test Symposium, Montreal, Canada, 2000.
- [14] Gaitanis, N.: "Design of totally self-checking TMR fault-tolerant systems", In IEEE Transactions on Computers, 1988.
- [15] Semonov, O., Vassighi, A., Sachdev, M.: "Impact of Self-Heating Effect on Long-Term Reliability and Performance Degradation in CMOS Circuits", in IEEE Trans. on Device and Materials Reliability, 2006.
- [16] Srinivasan, J., Adve, S.V., Bose, P., Rivers, J., Hu, C.-K.: "RAMP: A Model for Reliability Aware Microprocessor Design", In IBM Research Report, RC23048, 2003.
- [17] Chandrakasan, A.P., Sheng, S., Brodersen, R.W.: "Low-power CMOS digital design", In IEEE Journal of Solid-State Circuits, 1992.
- [18] Zhu, D., Melhem, R., Mossé, D., Elnozahy, E.: "Analysis of an energy efficient optimistic TMR scheme", In Proc. of International Conference on Parallel and Distributed Systems, Newport Beach, USA, 2004.
- [19] Elnozahy, e., Melhem, R., Mossé, D.: "Energy-efficient duplex and tmr real-time systems", In Proc. of Real-Time Systems Symposium, Austin, USA, 2002.
- [20] Sämrow, H., Cornelius, C., Salzmann, J., Tockhorn, A., Timmermann, D.: "Utilizing Parallelism of TMR to Enhance Power Efficiency of Reliable ASIC Designs", In Proc. of International Conference on Computer Engineering and Systems, Cairo, Egypt, 2010.
- [21] Koren, I., Krishna, C.: "Fault-tolerant Systems", M. Kaufmann, 2007.