Extending Lifetime Reliability Model for Multi-Threaded Architectures

Harini Sriraman
and
Pattabiraman Venkatasubbu
Vellore Institute of Technology, Chennai, India

As the processor technology points scale down, the hardware reliability of the processor, due to aging emerges as a significant design constraint. Currently it is estimated that the aging servers needs to be replaced ideally every three years in a data center kind of environment. This incurs huge cost. Under this scenario, it is very essential to explore methods to delay aging of processor cores. In this paper, we analytically establish the relation between life-time of the processor core and its multi-threaded workload. This paper answers the question 'how much multi-threading for how much reduction in aging?'. To answer the above question, we propose an analytical model to extend the relation between multi-threading and failure rate of processor core. To measure the delay in aging of processor, we use Aging Factor that is derived in our analytical model. We analyze the aging factor of the processor core at the granularity of structural units for different applications. Based on the proposed analytical model, a software tool AgeEstimate is designed that will take as input, power, temperature values for single threaded environment and estimate aging factor for multi-threaded environment. The results obtained from AgeEstimate for ALPHA based out-of order processor core are analyzed in this paper. With multi-threading, the instantaneous Mean Time to Fail (MTTF) for a workload can reduce up to 3%. Considering a base MTTF of 1 billion hours, the improvement due to multi-threading will be around $3 * 10^7$ hours.

Keywords: Multi-Threading, Parallelism, Aging of Processors, Processor Lifetime reliability, Hardware Fault

1. INTRODUCTION

Ensuring lifetime reliability of processor cores especially after relentless technology scaling is very important. Reliability of the processor core and its structural units are expressed in terms of Mean time To Failure (MTTF). In existing reliability models, MTTF for various fault mechanisms are calculated using voltage, power and activity factor. In recent technology points, lifetime reliability has become a major design constraint that limits the performance of the system. To satisfy this lifetime reliability constraint, as given in (Zu, Lefurgy, Leng, Halpern, Floyd, and Reddi, 2015), traditional guard banding wastes a lot of energy. In this paper we analyze the use of existing multi-threaded programming techniques for improving reliability measures in a processor core. By proper use of program constructs, reliability values can be improved.

1.1 Contribution

A Analytical Model to relate lifetime reliability of processor cores and multithreading

B Analysis of the proposed model using simulated reliability values (aging factor) for an out of order processor core

2. RELATED WORK

In recent times, while the power and temperature simulators have been extended to accommodate changes in the design and fabrication of recent technology cores, reliability simulators has to be explored in this area. In the recent years, there has been a significant increase in work done to model accurately the energy consumption in modern processor technologies. One of the major influences to this power modeling is the use of SMT and its effect. To this extent, temperature modeling has also recently included the multi-threading effects on steady state operating temperature of processor cores. While power and temperature modeling has included multi-threading effects, lifetime reliability model of processor cores has not completely included this for accurate measurement of reliability values. In this paper, we extend literature to emphasize the importance of including the process parameters for wear-out reliability modeling of processor cores.

2.1 Existing Reliability Models

Existing models, (Huang, Rajamani, Stan, and Skadron, 2011), (Chakraborty and Pan, 2013), (Sorin, 2009), (Hari, Li, Ramachandran, Choi, and Adve, 2009) and (Gizopoulos, Psarakis, Adve, Ramachandran, Hari, Sorin, Meixner, Biswas, and Vera, 2011), uses MTTF as reliability measure for processor cores and its structural units. There are lifetime reliability models that models wear-out faults due to failure mechanisms like Electron Migration (EM), Stress Migration(SM).Time Dependent Di-electric Breakdown (TDDB), Thermal cycling (TC) and Negative Bias Temperature Instability (NBTI). In this paper, we analyze the effect multi-threading on the reliability values. Table 1 lists the advancements in power and thermal models with respect to multi-threading.

Table I: Recent Advancements in Power and Thermal Models

Existing Literature	**Contribution**
Energy modelling for hardware multi- threaded processor by (Huang et al., 2011) and (Brooks, Dick, Joseph, and Shang, 2007)	—Considers Energy as integration of power over time —Power dissipation is due to base cost and thread cost —The power increase is less significant once the pipeline is full
Power consumption for multi-threaded workloads (Huang, 2011), (Thiyagalingam and Trefethen, 2014) and (Tang, Yang, Lee, and Jha, 2015)	—Power consumption for single thread workload and multi-thread workload are analysed —Power equation using 'n' threads can be given in terms of power using single thread as follows $P_n = \gamma P_1$ where $\gamma > 1$
Thermal Models based on Workload Phase Detection in multi-core processors (Liao, Hsieh, and Lee, 2017), (Fadishei, Deldari, and Naghibzadeh, 2014) and (Saravanan, Chandran, Punnekkat, and Kothari, 2011)	—Models temperature of a particular node as a linear combination of current node temperature, idle temperature and variation in temperature —Thermal variation with respect to workload characteristics modelled

3. EXTENDING THE METRIC FOR ASSESSING LIFETIME RELIABILITY OF A MULTI-THREADED SYSTEM

To quantify the gravity of reliability issues as technology scales, MTTF values for different applications from 45nm to 22nm are estimated. For this estimation, the splash2 benchmark applications are executed on the ALPHA based out-of order processor architecture of GEM5 performance simulator. The performance metrics are then given to McPAT power simulator with technology node parameter to be set as 45nm. The corresponding power trace is applied to the HotSpot6.0 with a floor plan of 45nm technology architecture that resembles ALPHA processor 21264. The power and temperature trace values are fed to the RAMP2 simulator and the MTTF values are estimated. To obtain the reliability value of 45nm technology node, the scaling parameters are fixed as shown in Table 2.

Table II: Scaling Factors for MTTF Calculation Beyond 22nm

Technology Node	**Relative Area**	**Relative Capacitance**	**Frequency** (GHz)	**Power Density** (W/mm^2)	**Vdd** (V)	**Tox** (A)
45nm	1	1.0	1.9	1.0	1.1	1.5
32nm	0.62	0.7	2	1.2	1	1.4
22nm	0.62	0.65	2.4	2.1	0.8	1.2
14nm	0.46	0.65	3.1	3.09	0.7	1
10nm	0.43	0.6	2.5	3.64	0.6	0.84

Failure Mechanisms considered for calculating MTTF includes Electron Migration (EM), Stress Migration (SM), Time Dependent Dielectric Break Down (TDDB) and Negative Bias Temperature Instability (NBTI). The same process is carried out for 32nm and 22nm technologies to derive the MTTF values. The power and temperature simulator works till 22nm technology. So in order to estimate the values beyond 22nm, i.e., 14nm and 10nm technologies, we use the scaling factors as given in [21]. The reliability values thus obtained for each bench mark application averaged over threads is plotted in Figure 1. An analytical model to relate multi-threading with lifetime reliability is discussed in Section 3.2. From Fig 1, it can be seen that the time to fail decreases with technology scaling. Hence it is very essential with aggressive technology scaling to take counter measures to aging.

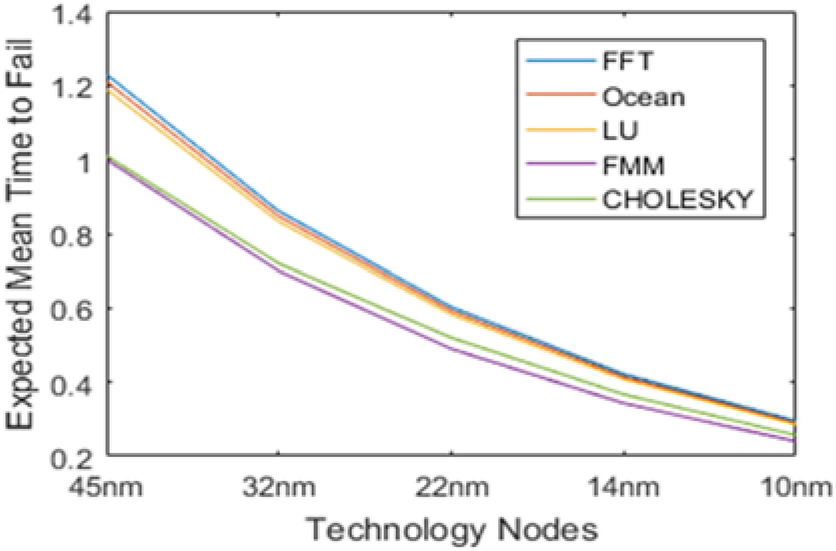

Figure 1. Technology Scaling versus MTTF Scaling

To derive an analytical relation between multithreading and lifetime reliability, the relation between multi-threading and power, temperature has to be understood. From the recent literatures like (Thiyagalingam and Trefethen, 2014), (Huang, 2011) this relation is made evident. The relation is explained below.

—Instantaneous power value increases

—Execution time decreases

—Overall energy consumption decreases

—Temperature instantaneous increases

The analytical model proposed in section 3.2 is based on the system model explained in section 3.1.

3.1 System Model

The failure rate of the processor has been calculated by integrating the failure rates arising as a result of various workloads of the individual structures over time. Here time varies from 0 to ∞. For the purpose of calculating the aging factor, time ‘t’ is divided in to intervals τ_{j} where τ_{j} is the execution time of each workload. The properties of τ_{j} are as given in equations 1 and 2.The system model is better understood with Fig 2.

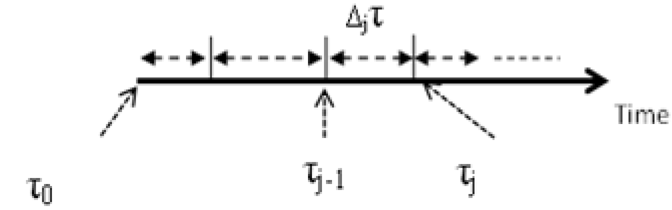

Figure 2. Time Divided in to τ_j

$$\tau_j < \tau < \tau_{j+1} \tag{1}$$

$$\tau_j = \Delta_{j\tau} + \tau_{j-1} \tag{2}$$

From both the above equations it could be seen that τ is cumulative. The failure distribution considered in this model is Weibull distribution since it is the best suited distribution model for lifetime distribution of processors. Conventions used in the proposed analytical model are listed in Table 3.

Table III: Conventions Used

Convention	**Explanation**
$MTTF_{Overall}$	Overall MTTF Across Applications
FIT	Fault In Time
τ_{jm}	Time at instance ’j’ with ’m’ threads
E_1	Energy consumption for ’1’ thread
E_n	Energy consumption for ’n’ threads
P_1	Power consumption for ’1’ thread
P_n	Power consumption for ’n’ threads
Γ	Ratio between P_n and P_1
β	Ratio between E_n and E_1
T_j	Instantaneous Temperature
s_j	States at instance ’j’
C_j	Aging rate at instance ’j’
$\Delta_{jm}\tau$	Execution time of application with ’m’ threads at instance ’j’
Θ	Scaling Parameter
$\theta(T_j, s_j)$	Instantaneous MTTF

3.2 Mathematical Model to Relate Multi-Threading with Aging Rate

For multithreaded programming, based on (Kerrison and Eder, 2015), the following equations can be derived.

$$\Delta_{j1\tau} = \alpha\Delta_{jn\tau} \tag{3}$$

$$P_n = \gamma P_1 \tag{4}$$

$$E_n = \beta E_1 \tag{5}$$

$$\theta(T_{j1}, s_{j1}) = \delta\theta(T_{jn}, s_{jn}) \tag{6}$$

For the best parallel application the values of α, γ and β are n, n and 1 respectively. Like power, instantaneous MTTF decreases with increase in number of threads. For an application that cannot be parallelized completely and has many depending constructs in the logic, the values of α and γ are much less than 'n'. The value of β depends on the ratio of $\frac{\alpha}{\gamma}$.

3.3 Relation between Multi-Thread and Instantaneous MTTF

In this section, the relation between multi-thread and instantaneous MTTF for Electro Migration has been established. The same can be applied to other failure mechanisms as well. Instantaneous MTTF for EM in general is given in equation 7. This can be rewritten for multiple threads, namely, '1' thread and 'n' threads as in equations 8 and 9 respectively. Ratio of equation 8 to equation 9 is calculated.

$$\theta(T_{jm}, s_{jm})_{EM} = A0\frac{(e^{E_{a_{EM}}/kT})}{V^s f^s P^s} \tag{7}$$

$$\theta(T_{j1}, s_{j1})_{EM} = A0\frac{(e^{E_{a_{EM}}/kT})}{V^s f^s {P_1}^s} \tag{8}$$

$$\theta(T_{jn}, s_{jn})_{EM} = A0\frac{(e^{E_{a_{EM}}/kT})}{V^s f^s {P_n}^s} \tag{9}$$

3.3.1 *Aging Factor with respect to Multi-Threading.* Now considering the reliability measures given by (Huang, 2011), the following equations are written.

$$C_j = C_{j-1} + \frac{\theta}{\theta(T_j, s_j)}(\tau_j - \tau_{j-1}) \tag{10}$$

Here C_j is the aging rate of components in a processor at instance 'j' , Θ is the scaling parameter and $\theta(T_j, s_j)$ is the instantaneous MTTF value calculated using black's equation. Now rewriting equation 10 with respect to multi-threading, equations 11 and 12 are arrived at. Equation 11 is the aging rate with '1' thread and equation 12 is the aging rate with 'n' threads. Rearranging equations 11 and 12, 13 and 14 are derived.

$$C_{j1} = C_{j1-1} + \frac{\theta}{\theta(T_j1, s_j1)}(\tau_j1 - \tau_{j1-1}) \tag{11}$$

$$C_{jn} = C_{jn-1} + \frac{\theta}{\theta(T_jn, s_jn)}(\tau_jn - \tau_{jn-1}) \tag{12}$$

$$C_{j1} - C_{j1-1} = \frac{\theta}{\theta(T_j1, s_j1)}(\tau_j1 - \tau_{j1-1}) \tag{13}$$

$$C_{jn} - C_{jn-1} = \frac{\theta}{\theta(T_jn, s_jn)}(\tau_jn - \tau_{jn-1}) \tag{14}$$

Here $C_{jm} - Cjm - 1$ is the aging factor at instance 'j' since Cjm is cumulative where 'm' is the number of threads. Let C_1 and C_n denote the aging factor at instance 'j' for '1' and 'n' threads. C_1 and C_n are represented in equations 15 and 16 respectively. Ratio of equation 15 and equation 16 is given in equation 17. Simplifying equation 17 yields equation 19.

$$C_1 = \frac{\theta}{\theta(T_j1, s_j1)}(\tau_j1 - \tau_{j1-1}) \tag{15}$$

$$C_n = \frac{\theta}{\theta(T_jn, s_jn)}(\tau_jn - \tau_{jn-1}) \tag{16}$$

$$\frac{C_n}{C_1} = \frac{\frac{\theta}{\theta(T_jn, s_jn)}(\tau_jn - \tau_{jn-1})}{\frac{\theta}{\theta(T_j1, s_j1)}(\tau_j1 - \tau_{j1-1})} \tag{17}$$

$$\frac{C_n}{C_1} = \frac{\frac{\theta}{\theta(T_jn, s_jn)}(\tau_jn - \tau_{jn-1})}{\frac{\theta}{\theta(T_j1, s_j1)}(\tau_j1 - \tau_{j1-1})} \tag{18}$$

$$\frac{C_n}{C_1} = \frac{\delta\tau_{jn} - \tau_{jn-1}}{\tau_{j1} - \tau_{j1-1}} \tag{19}$$

$$\frac{C_n}{C_1} = \frac{\delta\Delta_{jn\tau}}{\Delta_{j1\tau}} \tag{20}$$

Equation 20 shows the ratio of the aging factor (that determines the scaling of the distribution) as directly proportional to instantaneous power ratio and inversely proportional to execution time ratio with respect to '1' and 'n' threads.

For the best case scenario, the overall MTTF improves with increase in number of threads. This can be proved with the lemmas

Lemma 1. *For parallel applications whose performance improve proportional to the number of threads, instantaneous MTTF decreases with increase in number of threads.*

$$\delta < 1$$

Proof. Consider a parallel application that is executed with a single thread and multiple threads respectively. From the article by (Zu et al., 2015), it is clearly given that, for a highly parallelizable application, the values of α, γ and β will be 'n', 'n' and '1' respectively. Substituting these values in equations 3, 4 and 5, the following derivation can be made.

$$\begin{aligned} \frac{1}{\delta} &= (\frac{E_n}{\tau_n})^s (\frac{\tau_1}{E_1})^s \\ &= n \\ \delta &= \frac{1}{n} \end{aligned}$$

For multi-threaded applications, $n > 1$. Hence proved.

Lemma 2. *For parallel applications whose performance improve proportional to the number of threads, overall MTTF increases with increase in number of threads.*

$$C_{j1} > C_{jn}$$

$$\frac{C_{jn}}{C_{j1}} < 1$$

Proof. Substituting the values of α, γ and β as 'n', 'n' and '1' respectively, aging factor can be derived as given below.

$$\begin{aligned}\frac{C_n}{C_1} &= \frac{\Delta_{jn}\tau}{n\Delta_{jn}\tau}\\ &= \frac{1}{n}\\ &< 1\end{aligned}$$

The proof here is given for the best cases of parallel applications. The above theorem can be extended for other types of applications. Hence for parallelizable applications, the aging factor will be inversely proportional to the number of threads.

4. AGE ESTIMATE

Based on the above analytical model, a tool called AgeEstimate is written to estimate the aging pattern of a given application when executed with 1 to 'n' threads. AgeEstimate is written using python script and the overall work flow of the tool is given in Fig 3.

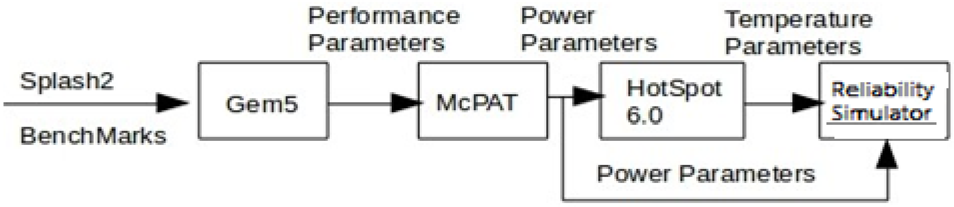

Figure 3. Overall Simulation Flow

AgeEstimate works along with power, temperature and performance simulators. For a particular application, the performance, power and thermal parameters for a single threaded environment is the input of AgeEstimate. This tool will calculate instantaneous MTTFs based on blacks equation for four reliability mechanisms namely, EM, SM, TDDB and NBTI. The execution time of the applications are extracted from performance simulator. Using the above values, aging rate at jth instance as given in equation 11 is calculated by fixing the shape parameter of weibull distribution (Huang, 2011) to be '4.0'. This calculation is for aging rate for single thread. To calculate multi-threaded aging rate, the values of τ_{jn} is derived by considering a speed up proportional to the number of threads. The aging factor as explained in equation 15 is finally calculated. Hence the output of AgeEstimate will show the delay in aging of processor cores for the workloads executed, when varying the number of threads. Currently we developed this tool for parallel applications that will result in speedups proportional to the number of threads. This can be extended for other cases by considering different speed ups that can be given as input to the tool. Autonomously looking at the code, finding the speed up and aging factor can be carried out in the future. In section 5, the experimental setup of the above work is explained in detail.

5. EXPERIMENTAL SETUP

Simulation setup used for carrying out the experimental analysis consists of a performance simulator (Gem5), a power simulator (McPAT),a thermal simulator (HotSpot6.0), a reliability simulator, RAMP 2.0 by (Srinivasan, 2006) and the proposed AgeEstimate calculation. First the instantaneous MTTF values were simulated for 45nm to 22nm for each individual structure based on the work flow. Then those values were used for the analytical examination of the aging factor calculation given in equation 20. For simulation purposes, a hypothetical ALPHA based processor architecture was considered for the analysis of its effect on delaying the aging effect. The floor plan of the simulated hypothetical architecture at 45nm technology is shown in Figure 4. The base processor details are given in Table 4. The flow of data between multiple simulators used is shown in Fig. 0**??**. The statuses of the shared units in the hypothetical architecture are shown in Table 5. The analysis of the aging factor results are detailed in section 6.

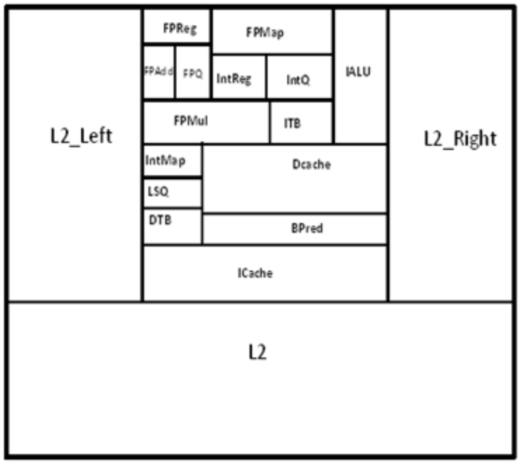

Figure 4. Floor Plan of Simulated Architecture at 45nm

Gem5 simulator is configured to simulate an hypothetical out-of order processor that loosely resembles ALPHA 21264. Splash2 benchmarks (FFT, OCEAN, LU, CHOLESKY, FMM) parallelized using Pthreads are compiled with ALPHA cross compiler and the binaries are loaded in to the full system simulator option of Gem5. Temperature, power and reliability values are then obtained for the corresponding simulators.

6. RESULTS AND DISCUSSIONS

For experimental evaluation purpose, we considered splash2 benchmarks whose speed up for different applications is proportional to the number of threads. The performance values are obtained for splash2 benchmarks on the hypothetical architecture using gem5. The temperature, power and reliability values are measured for 45nm, 32nm and 22nm technology node by setting appropriate values in the power, temperature tools. Instantaneous MTTF values are calculated by modifying the constant values of RAMP 2.0 for different technology nodes from 45nm to 22nm as given in Table 2. Average of δ values across technology nodes is found for different number of

Table IV: Details of Base Processor Features

Base Processor Feature	**Value**
Process Technology	45nm
Base Temperature	380K
Interconnection Projection	Aggressive Wire
Issue Width	4
Commit Width	4
Number of Hardware Threads	1 to 4
Branch Predictor	Bi-mode, Local BP, Tournament
Icache, Dcache Capacity	65536
DTB, ITB	128 entries
BTB Capacity	6144
Clock Rate	2000MHz
Machine Bits	64
Power Gating	Not Enabled
Virtual Address Width	64
Physical Address Width	64
ROB Size	80

Table V: Shared Status of Units

Private	**Shared**	**Partitioned and Tagged**
IBuffer	Icache, IALU, FPU, Dcache, Register Files	BTB, Branch Predictor, Instruction Decoder, ITB, DTB, LdStQ

threads. As given in the analytical explanation, the values comes around '1'. This is described in Figure 5.

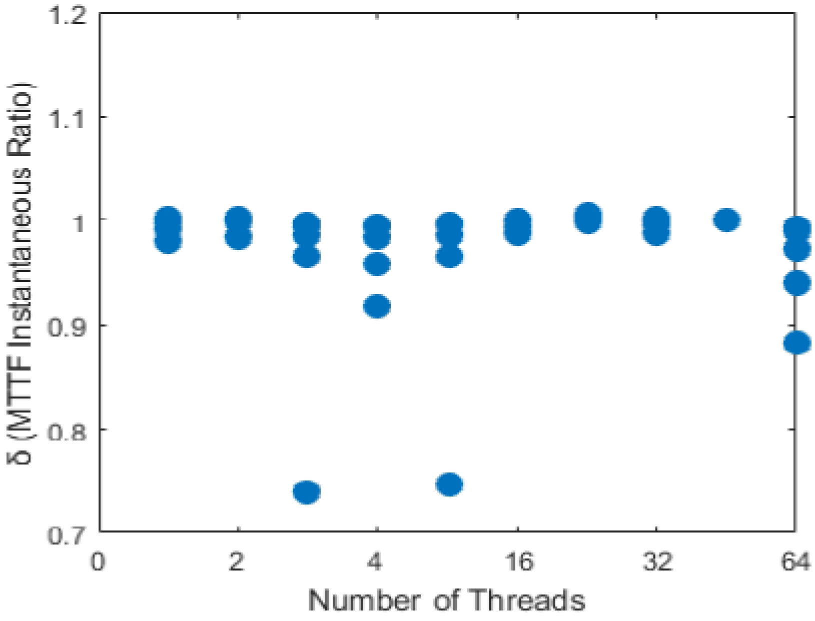

Figure 5. Instantaneous MTTF Ratio

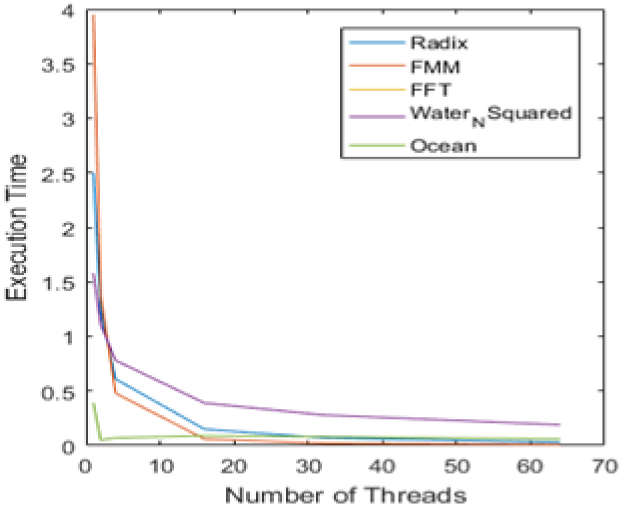

Figure 6. Multi-Threading Versus Execution-Time

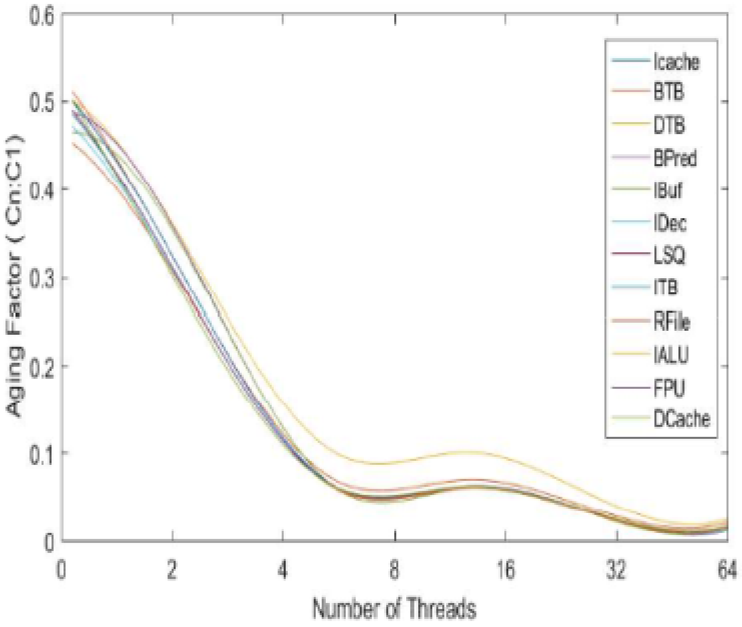

Figure 7. Analytical Aging Factor

To understand the effect of multi-threading on aging factor in a real time out-of order processor, we considered ALPHA based processor described in gem5. The status of the shared units in the processor is given in Table V. With respect to this out-of order processor core, the values of temperature, power and reliability are obtained as before. The values were then substituted in equations 11, 12 and the aging factor is calculated. The results are given in Figure 8 and Figure 9. As can be seen, due to the shared status, some of the units in the architecture see diminishment with respect to aging factor while the other units see improvement. The diminishment and improvement of the corresponding units in percentage is given in Figure 10 and 11 respectively.

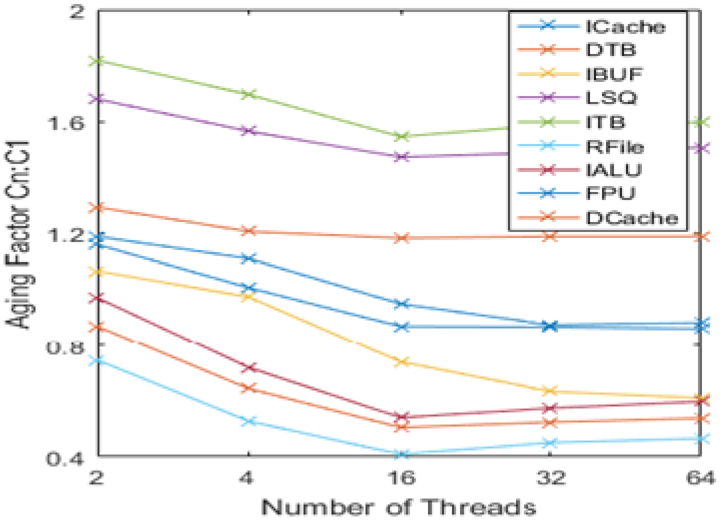

Figure 8. Private Unit's Aging Factor

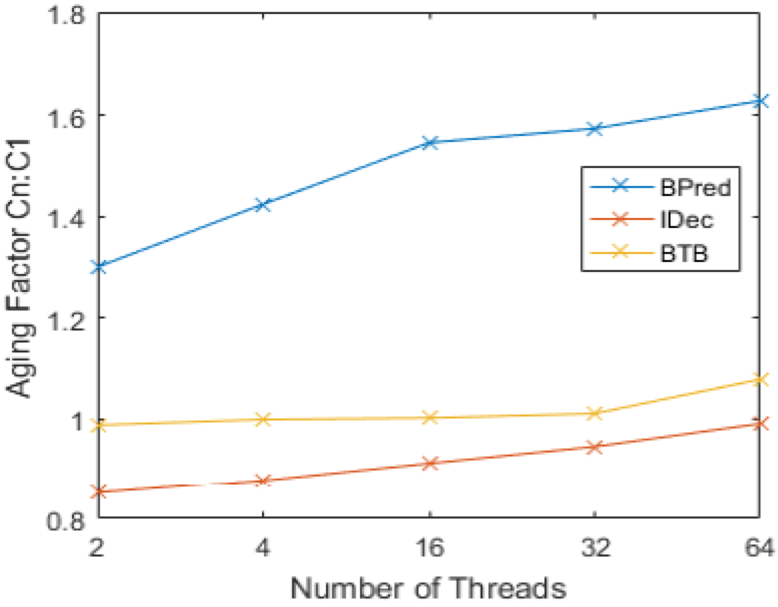

Figure 9. Shared Unit's Aging Factor

Comparing Figures 7 and 8, analytical improvement in aging factor is much more compared to its simulated counterpart. This could be attributed to the following factors.

—Even though the applications are parallel, the processor architecture's simple pre-fetch, memory system organization and shared units prevents the application to fully utilize the benefits of parallelism on aging factor

From Fig 9, we see some of the shared units show aggravated aging factor. The amount of improvement and diminishment is shown in Figures 10 and 11 respectively. The solution for the above problem is to provide redundant units or spare units that could be used so that reliability can be maintained. Even though some of the units show negative impact when using

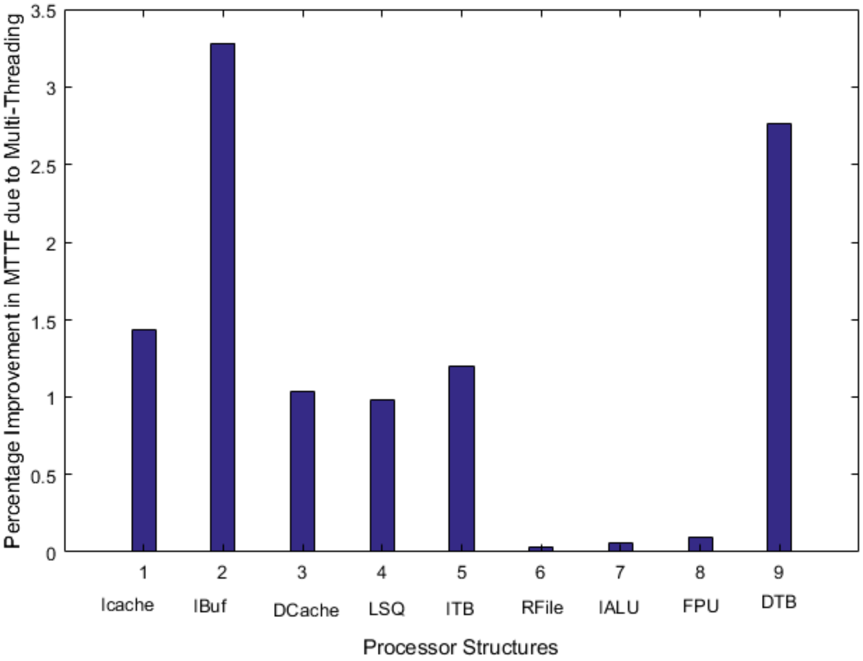

Figure 10. Improvement due to Multi-threading

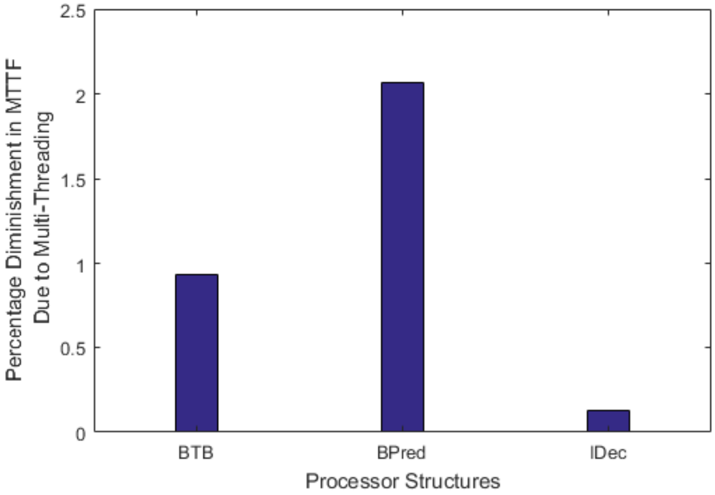

Figure 11. Diminishment due to Multi-threading

multi-threading, the overall life time of the processor will improve. This is due to the fact that majority of the units benefit from multi-threading. Now averaging the aging factors across multiple threads for different applications and extending them analytically for technology nodes till 10nm, we see there is definite improvement as shown in Figure 12 when compared to Fig 2.

7. CONCLUSION AND FUTURE WORK

The major contribution of this work is to extend the relation between multi-threading and aging rate of processor structures. As could be seen from the analysis, proper use of multi-threading will

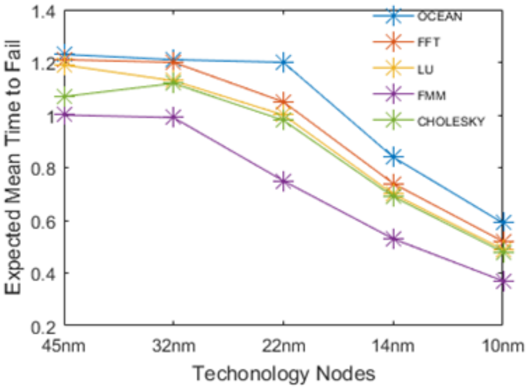

Figure 12. Technology Scaling Versus Multi-Threading

delay the aging process. From the analysis, it is shown that aging factor reduces proportional to number of threads for highly parallel applications. The proposed analysis is useful to accurately measure reliability values according to its workload conditions. Based on this analysis, the following changes can be made to the micro architecture for efficiency and reliability.

—Provide redundant units instead of shared units for highly utilized units

—Provide flexible guard band depending on the aging factor so that maximum efficiency can be achieved

Currently the tool proposed works for the hypothetical ALPHA based architecture. In future this could be extended for multiple architectures. The tool proposed has the assumption that the applications are highly parallel and the speed up is proportional to the increasing number of threads. This may not be true for all applications. In future this could be made to include many different speedups and project aging factor for them. Since this is the first of its kind to relate multi-threading with aging, a lot of enhancements can be made in the future.

References

Brooks, D., Dick, R. P., Joseph, R., and Shang, L. 2007. Power, thermal, and reliability modeling in nanometer-scale microprocessors. *Ieee Micro 27,* 3.

Chakraborty, A. and Pan, D. Z. 2013. Skew management of nbti impacted gated clock trees. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 32,* 6, 918–927.

Fadishei, H., Deldari, H., and Naghibzadeh, M. 2014. Pre-execution power consumption prediction of computational multithreaded workloads. *Cluster computing 17,* 4, 1323–1333.

Gizopoulos, D., Psarakis, M., Adve, S. V., Ramachandran, P., Hari, S. K. S., Sorin, D., Meixner, A., Biswas, A., and Vera, X. 2011. Architectures for online error detection and recovery in multicore processors. In *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2011.* IEEE, 1–6.

Hari, S. K. S., Li, M.-L., Ramachandran, P., Choi, B., and Adve, S. V. 2009. mswat: low-cost hardware fault detection and diagnosis for multicore systems. In *Microarchitecture, 2009. MICRO-42. 42nd Annual IEEE/ACM International Symposium on.* IEEE, 122–132.

Huang, L. 2011. Lifetime reliability of multi-core systems: Modeling and applications. Ph.D. thesis, Chinese University of Hong Kong.

Huang, W., Rajamani, K., Stan, M. R., and Skadron, K. 2011. Scaling with design constraints: Predicting the future of big chips. *IEEE Micro 31,* 4, 16–29.

Kerrison, S. and Eder, K. 2015. Energy modeling of software for a hardware multi-threaded embedded microprocessor. *ACM Transactions on Embedded Computing Systems (TECS) 14,* 3, 56.

Liao, M.-H., Hsieh, C.-P., and Lee, C.-C. 2017. Systematic investigation of self-heating effect on cmos logic transistors from 20 to 5 nm technology nodes by experimental thermoelectric measurements and finite element modeling. *IEEE Transactions on Electron Devices 64,* 2, 646–648.

Saravanan, V., Chandran, S. K., Punnekkat, S., and Kothari, D. 2011. A study on factors influencing power consumption in multithreaded and multicore cpus. *WSEAS Transactions on Computers 10,* 3, 93–103.

Sorin, D. J. 2009. Fault tolerant computer architecture. *Synthesis Lectures on Computer Architecture 4,* 1, 1–104.

Srinivasan, J. 2006. Lifetime reliability aware microprocessors. Tech. rep.

Tang, A., Yang, Y., Lee, C.-Y., and Jha, N. K. 2015. Mcpat-pvt: Delay and power modeling framework for finfet processor architectures under pvt variations. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems 23,* 9, 1616–1627.

Thiyagalingam, J. and Trefethen, A. E. 2014. Understanding the interactions hardware/software parameters on the energy consumption of multi-threaded applications.

Zu, Y., Lefurgy, C. R., Leng, J., Halpern, M., Floyd, M. S., and Reddi, V. J. 2015. Adaptive guardband scheduling to improve system-level efficiency of the power7+. In *Proceedings of the 48th International Symposium on Microarchitecture.* ACM, 308–321.

Harini Sriraman completed her B.E from Madras University and M.E from College of Engineering, Guindy, Chennai, India. She has a total professional experience of more than 8 years in various institutions and industry. She has published articles in reputed international peer reviewed conferences and journals. She is currently working with Vellore Institute of Technology, Chennai, as Assistant Professor in the School of Computing Science and Engineering. Her teaching and research interest includes Parallel Computer Architectures, Fault Tolerant Systems, Parallel Programming and Reconfigurable Computing.

Dr. Pattabiraman Venkatasubbu obtained his Bachelor's from Madras University and Master's degree from Bharathidasan University. He completed his PhD from Bharathiar University, India. He has a total Professional experience of more than 16 years working in various prestigious institutions. He has published more than 30 papers in various National and International peer reviewed journals and conferences. He visited various countries namely few China, Singapore, Malaysia, Thailand and South Africa etc. for presenting his research contributions as well as to giving key note address. He is currently an Associate Professor and Program-Chair for Master's Programme at VIT University-Chennai Campus, India. His teaching and research expertise covers a wide range of subject area including Data Structures, Knowledge Discovery and Data mining, Database echnologies, Big Data Analytics, Networks and Information security etc.