

Reliability and Power Management of Integrated Systems

T. Simunic, K. Mihic and G. De Micheli

Abstract — In this work we present a new approach for dynamic reliability and power management of Integrated Systems, such as Systems on Chips (SoCs) and Networks on Chips (NoCs). With aggressive transistor scaling, decreased voltage margins, and increased processor power and temperature, reliability assessment has become a significant issue in design. Our work combines for the first time dynamic power management with a reliability models. The joint model is used to determine system level reliability as a function of failure rates, system configuration and power management policies. We show that the overall system reliability is strongly affected by reliability network topology and power management policy.

I. INTRODUCTION

Future integrated systems will be designed using many high-level components such as programmable cores. Advances in technology will lead to higher device density and operating frequency. At the same time, supply voltages will be reduced to curtail energy dissipation, with the unfortunate effect of reducing noise immunity. As a result, computation, storage and information transmission on chip will be subject to malfunctions, which may be the source of system-level failures. The international technology roadmap on semiconductors (ITRS) predicts that reliability requirements will become a significant design parameter in the next few years [15]. Thus, design for yield and reliability is becoming a very active area of research. Several design styles address reliability issues. When considering reliable interconnect requirements, Networks on Chips (NoCs) (also called micro-networks) can provide an effective backbone for supporting standby components, which can be used to enhance system-level reliability. At the same time, the network itself can be made highly reliable, by using encoding and packetization techniques [2-6].

Dynamic power management (DPM) has been applied, in various forms, to both single and networked components [13,16]. Reducing energy consumption to the required levels ensures correct and useful operation of the integrated systems. DPM also affects the reliability of the system components. Lowering power consumption helps reduce the overall chip temperature, but on the other hand it can increase the probability of data errors. In addition, frequent core transitions between active and low power states can cause a decrease in the overall core reliability. As a result, there is a

need to evaluate the SoC/NoC reliability along with power consumption and performance. There are several interesting problems that can be considered.

The first one is to analyze system-level reliability as a function of time, for a given reliability topology, components and DPM policy. This analysis allows us to determine whether the effects of DPM are beneficial for reliability, and in particular if such benefits are long or short term. The second problem is to incorporate reliability as an objective into DPM policy optimization. In other words, the goal would be to reduce energy consumption and enhance reliability. This assumes again a fixed reliability topology and a fixed set of components. A third set of problems include the choice of components and topologies to achieve reliable low-energy design. Note that while the previous problem addressed only run-time strategies, this problem involves now also system design issues.

In this paper we focus on the first problem as an enabler to understanding the relationship between run-time power management and reliability analysis. We study reliability, performance and power consumption of SoCs/NoCs by modeling system level reliability as a function of failure rates, system configuration and DPM policies. The overall objective is to be able to introduce design constraints, such as *mean time to failure* (MTTF), in the design space spanned by performance and energy consumption.

The rest of the paper is organized as follows. Section II discusses related work. We introduce our approach for assessing reliability and power consumption in Section III. A simulation methodology we developed is presented in Section IV, followed by the study of reliability and power consumption tradeoff as a function of various system parameters in Section V. Finally, Section VI summarizes the contributions of our work.

II. RELATED WORK

Integrated systems have been in production for a while in the form of Systems on a Chip (SoCs). A number of issues related to SoC design have been discussed to date, ranging from managing power consumption (for an overview see [2]), to addressing problems with interconnect design (e.g. AMBA and CoreConnect standards [27,28]). Design of Networks on Chips is a relatively new field with numerous challenges. Recent research results in the area of NOC design and optimizations is given in [2-6]. There are a few NOC case

studies that have been presented recently [25,26]. Another interesting example is Maia processor [9] which consists of 21 satellite units connected via two-level hierarchical reconfigurable network. Large energy savings were observed due to the ability of Maia to reconfigure itself according to application needs. Reduction of energy consumption in NOCs is challenge that needs to be considered, in tandem with the design of the on-chip communication network. Power savings obtained by only scaling down supply voltage levels are not going to be sufficient to compensate for a higher complexity, a larger interconnect capacitance and resistance, a higher operating frequency and an increased gate leakage [7]. Previous work for energy management of NOCs mainly focused on controlling the power consumption of interconnects [2-6,8], while neglecting managing power of the cores. An outline of possible approaches for energy savings in NOC cores is presented in [2]. A stochastic optimization methodology for core-level dynamic voltage and power management of NOCs with both node and network centric views using a closed-loop control model has been presented in [13].

A good summary of research contributions that combine performance and reliability measures is given in [10]. Microarchitectural level reliability work on soft errors is described in [11]. Another way to improve system reliability and increase processor lifetime is by implementing redundancy at the architecture level, as discussed in [12]. A number of fault-tolerant microarchitectures have been proposed that can handle hard failures at performance cost [17,18] and area cost [19]. RAMP models chip wide mean time to failure as a function of the failure rates of individual structures on chip due to different failure mechanisms [20]. It can be combined with architecture-level simulators that give power and temperature estimates needed by the reliability models. Minimizing energy and performance by exploiting architecture and application-level adaptability has been presented in [21,22,23]. The work presented in [24] introduces Dynamic Fault-Tolerance Management (DFTM) improves system reliability due to soft failures with the particular attention to energy efficiency, computation performance and battery lifetime.

In contrast to previous work, our contribution presents the first ever combination of power management and reliability models for integrated systems (SoCs and NoCs). As reliability is strongly influenced by component power consumption, our approach will enable designers to obtain even more accurate estimates of the overall system energy consumption and long term reliability. We accomplish this goal by modeling the system level reliability as a function of failure rate, system configuration and DPM policies.

III. RELIABILITY AND DPM

Integrated systems can be abstracted by a reliability network, i.e., a connection of components labeled by their failure rates [20]. The network shows, by means of

series/parallel connection of components, the conjunctive/disjunctive relations among component operating state to insure system correct operation. Failure rates, defined as the speed at which components are likely to fail, in many cases depend on the operation state of a component, as when DPM is applied. Our objective is to evaluate system-level reliability as a function of time. This problem is solved by means of simulation rather than analytical techniques since future integrated systems might be very large NoCs with arbitrary topologies. The simulator incorporates the notion of power-manageable components and emulates a DPM policy. To the best of our knowledge this is the first time that reliability measures have been modeled jointly with DPM. We first present a reliability model and then we outline the DPM model.

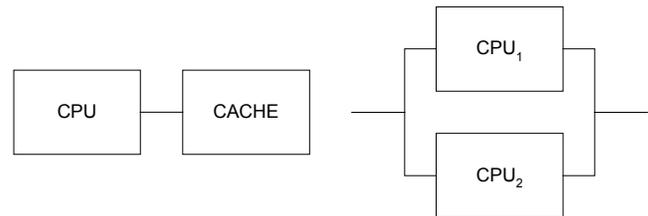


Figure 1. Series and parallel reliability systems

Integrated system consist of computational, storage and communication resources. The reliability network of a SoC/NoC is a graph, where resources are modeled as nodes, and their functional relation are expressed as edges. The network expresses conjunctive and disjunctive requirement for the system to work properly, based on the status of its component resources. Figure 1 shows two simple examples. The first example is a reliability network of two components in series, for example a processing core and its cache. In order to have correct system operation, both components need to be working properly. The second example shows a parallel combination of two processors integrated on one core. In this particular system it is enough to have one of the cores operating correctly in order to have the correct system behavior.

Four main issues have been identified in [20] that directly impact reliability in NOCs:

1. increase in core temperature as a result of higher power density causes exponential raise in failure rate
2. decrease in minimum feature size causes higher failures due to dielectric breakdown, interconnect wear-out and higher leakage power which in turn causes faster thermal breakdown
3. higher number of transistors integrated on the die implies a higher chance of failure of any one of the individual devices
4. wide spread power management helps reduce temperature, but at the same time introduces power cycling which can cause a decrease in reliability.

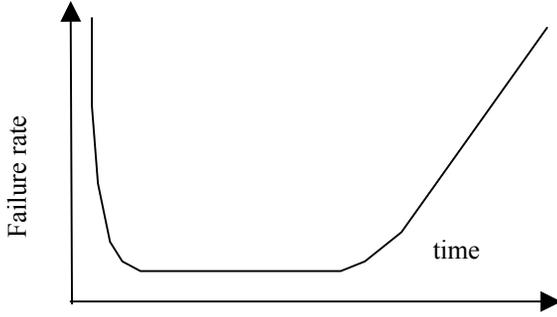


Figure 2. Bathtub curve

In general, failure rates are dependent on aging and on temperature. The first effect is shown clearly by the bathtub curve, shown in Figure 2. The curve has three distinctly different regions. Initial burn-in period and final wear-out period are typically modeled with Weibull distribution, while the failures during useful life are best described using an exponential distribution with a constant failure rate. Since we are interested in assessing the reliability over a typical operation time, we assume that the failure rate is constant in time, as shown by the middle range of the curve. Clearly, the value of failure rate is a function of many parameters, some of which are temperature, power state of the component and the frequency of switching between power states. Thus any given component will have multiple failure rates that describe it.

The reliability of a system is the probability function $R(t)$, defined on the interval $[0, \infty]$, that a system will operate correctly with no repair up to time t . Another variable commonly used to describe the system reliability characteristic is mean time to failure (MTTF) as shown in Equation 1.

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

Because we use a constant failure-rate model in any one of the component states, we can model the component reliability using exponential distribution with system failure rate, λ_f , as shown below. Mean time to failure is then defined as $1/\lambda_f$.

$$R(t) = e^{-\lambda_f t} \quad (2)$$

Integrated systems, such as SoCs and NoCs, consist of many cores connected with a complex interconnect structure. Often when a core or interconnect fails, another core or interconnect take over its functionality. Thus such a system has built-in redundancy. In order to model the overall system reliability, we need to define the relationship between topology, redundancy and component power state.

The system components can be organized in a series and/or in parallel as shown in Figure 1. Overall system reliability can be calculated by applying rules for series and parallel reliabilities as needed.

$$R_{system}(t) = \prod_{i=0}^n R_i(t) \Rightarrow R_{system}(t) = e^{-\sum_{i=0}^n \lambda_{f_i} t} \quad (3)$$

The system built with n series components fails if any of its components fails (see above). For example, a processor consisting of a computational unit, storage and busses can be abstracted by a reliability network with three nodes connected "in series" since the joint correct functioning of the system depends on the correct operation of the three resources. The system failure rate is the sum of the failure rates of the three components assuming each component's rate is constant.

$$R_{system}(t) = 1 - \prod_{i=0}^n (1 - R_i(t)) \quad (4)$$

Alternatively, the parallel combination fails only if all n components that are in parallel fail (see above). Furthermore, not all parallel components have to be active at the same time, since reliable system operation depends on only one of them. The rest of the components can be in low power mode. When the currently active component fails, one of the redundant components transitions from low power into active mode. Thus we can both save power and improve system reliability, especially since failure rates for components in low power mode are lower than the rates for ones in the active state. The system reliability for such configuration, shown below, is a combination of active reliability rate, λ_f , and reliability rate of components that are standing by to the active component, λ_{sby} (note we assume in this equation all standby probabilities have the same rate). For example, a dual processor engine can be modeled by the "parallel connection" of two nodes, each abstracting a processor, when the operation of one processor suffices for the overall system to work.

$$R_{system}(t) = 1 - (1 - e^{-\lambda_f t})(1 - R_{sby}(t)) \quad (5)$$

$R_{sby}(t)$, as defined in Equation below, is the system reliability of $n-1$ components that are standing by one active component. The MTTF for a standby components is $(n-1)/\lambda_{sby}$, a factor of $(n-1)$ larger relative to MTTF for a single active component. Clearly, from reliability perspective it is very advantageous to have multiple redundant cores that are in standby until needed.

$$R_{sby}(t) = \sum_{s=0}^{n-1} \frac{1}{s!} (\lambda_{sby} t)^s e^{-\lambda_{sby} t} \quad (6)$$

System-level reliability is derived from the failure rate of its components. Failure due to four main mechanisms itemized above is modeled as a series combination, where the component failure rate is a sum of failure rates due to each failure mechanism. The detailed models for each failure mechanism are presented in [20]. Each of these rates depends on component temperature, and thus on component power consumption. As a result, we next discuss the modeling of

integrated system power and performance.

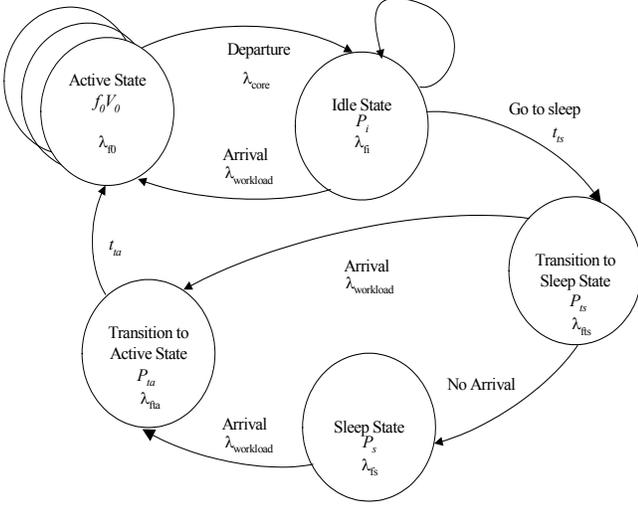


Figure 3. System Model

Power management can be done in various ways, and is applicable to computation, storage and communication resources. We model power management of each component by a *power state machine* (PSM), which is a state diagram relating service levels to the allowable transitions among them. Figure 3 shows a sample PSM for a core. Each state is labeled with a power consumption level and the appropriate failure rate (e.g. idle state failure rate in Figure 3 is λ_{fi} , while the power consumption is P_i). Obviously, components that are not power managed will show a failure rate that is independent of power states. Active state can be separated into multiple states differentiated by frequency and voltage of operation (e.g. f_0, V_0 , are equivalent to core processing rate λ_{core0} and the power consumption P_{a0}). Idle state represents mode in which core is active but not currently processing. Sleep state is a low power state the core can enter. Since the time and power consumption required to enter and exit sleep state is finite, we also model transition states. Each of the power states presented in the figure has a different failure rate, as each state represents different level of power consumption.

Since system reliability depends directly on the component power consumption and the frequency of entering low power states, it is important to model not just the power states, but also core workloads and power management policies implemented as a result of the workloads. The arcs on this graph represent possible transitions between the states with the associated transition times and rates (e.g. t_{ta} and λ_{core}). Table 1 summarizes all distributions used in modeling performance and power consumption. Workload follows exponential distribution with rate $\lambda_{workload}$ [14], much in the same way as reliability is modeled with exponential distribution. Similarly, cores processing rate is λ_{core} . The processing rate changes as the core's frequency of operation changes. Transition times to and from low-power states follow uniform distribution, such as the transition between active and sleep state shown in Figure 3, t_{ts} . The failure rates

change with each different low-power state since with lower power consumption comes also lower temperature and thus better failure rate. On the other hand, the failure rates worsen as the frequency of switching between power states increases [1,20] due thermal cycles introduced by different power consumption of each state. The level this thermal shock degradation is proportional to the temperature range, which is in turn, the function of voltage used by a core to accommodate different power states.

Table 1. Distribution summary

Component	State	Distribution	Parameters
Workload	Queue >=0	Exponential	$\lambda_{workload}$
Core	Active	Exponential	$\lambda_{core}, f-V \text{ curve}$
	Transition	Uniform	t_{min}, t_{max}

Power consumption is calculated based on the current power state of each component. The cores implement optimal power management policy presented in [14]. In the active state, the power manager decides only on the appropriate frequency and voltage setting, where as in the idle state the primary decision is which low-power state core should transition to and when the transition should occur. The next Section gives more details on how we implemented our simulation methodology.

IV. SIMULATION PLATFORM

As discussed in the previous Section, the overall system can be represented as a reliability network of PSMs. In general, analytical formulae can relate the reliability network topology and component failure rates to the system reliability. While analytical methods work well for smaller systems, larger systems with more complex topologies typically need to be evaluated using simulation. Advantages of simulation include, but are not limited to, handling general reliability network topologies, time-varying failure rates, as well as incorporating the effects of executing a DPM policy.

The simulator we built, as far as we know, is the first one that unifies power management with system level reliability model. The simulator is consisted of two tightly integrated components: a power management part that estimates and implements the optimum DPM policy, and a redundancy part that monitors and updates reliability network and returns the current reliability of the simulated system. Reliability network can be of any topology as long as it can be decomposed into series and parallel subsystem configurations. For now linked configurations are not supported. The simulator handles both active and standby redundancy models and alters the reliability network during the run time to accommodate for failed component(s). Reliability of a component is the function of time and failure rate, which in turn depends on stress time, frequency of changes of power states and voltage applied in different power states.

Each component in the system has a power manager that in turn consists of an estimator and a controller. The former estimates the parameters needed to recalculate optimal control

depending on the changes in the components' environment. The environment includes incoming traffic from the chip network, and special power management requests from other cores. The controller implements the optimal power management policy.

Simulating dynamic power management and reliability model with one tool enables us to observe correlation and dependency between power management and system level reliability. Results reported in the next Section highlight the strong relationship that exists between power management and system reliability.

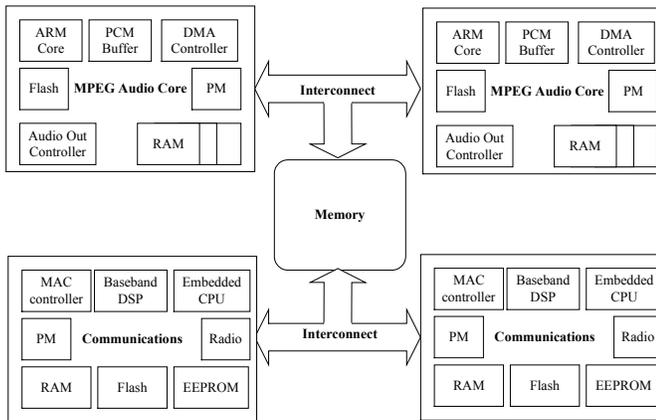


Figure 4. Basic Element Configuration

V. RESULTS

The methodology presented in this work was implemented for two integrated systems, a basic element of a larger system shown in Figure 4, and a larger system that consists of a few basic elements. The basic element, as shown in Figure 4, consists of four large cores, memory and interconnect network. In all our simulations we assure that the performance, as determined by the power management policy, does not change until the system fails due to reliability issues. Power and performance characteristics of each component are shown in Table 2. Three power states are supported by each core: active, idle, and sleep. The transition time from active to sleep and back to active state (shown in Table 2 as *A-S-A time*) is on the order of tens of milliseconds, which is slow enough to allow for dynamic parameter estimation and power management policy adaptation. Although memory normally supports multiple power states, in our simulation we focused on only active state in order to highlight the relationship between reliability and power management of the processing cores. In all our simulations the initial value of MTTF for each component is set to be 30 years, which is a typical value used in industry [20]. We use the acceleration factor to trade off the speed of the simulation with the accuracy in calculation of reliability. We found in our tests that acceleration factor value of 400 is appropriate. We validated our simulation results with analytical model for the example shown in Figure 4, with two redundant cores, and found simulation to be within 10% of the analytical results for

system reliability. Power management simulation results have been validated with measurements in [16].

Table 2. Core Specification

Specification	Audio1	Audio2	Com1	Com1	Mem	Net
Active P(mW)	700	700	1500	1500	2000	200
Idle P(mW)	216	216	1000	1000	NA	NA
Sleep P(mW)	0.3	0.3	100	100	NA	NA
A-S-A time(ms)	45.6	45.6	40	40	NA	NA

The simulation results shown in Figure 5 & 6 highlight the effect of power management on reliability of a redundant core due to different average sleep times and frequency of transitions between sleep states. On one hand longer sleep times both save power and improve reliability. On the other hand, frequent switching to and from sleep state causes a larger number of failures due to thermal cycling (TC). As feature sizes get smaller, TC failure mechanism becomes more dominant [29]. In both sets of simulations we use a basic element shown in Figure 4 with two out of four cores in redundant configuration. We compare reliability of a core that remains in standby mode with three cases where the redundant core is active: a) no power management (constant stress but no TC penalty), b) power management with low probability for transition to sleep mode, and c) power management with high probability for transition to sleep mode. Although the best results are always obtained when the core is in standby mode (not processing data), there are times when due to performance issues the redundant core has to remain active (processing data).

Figure 5 shows the reliability of a core when the effect of frequent switching of power states is negligible compared to improvements in failure rates gained by having a core in the sleep state. This type of results is more typical for larger feature sizes. Smaller feature sizes have significant degradation caused by large TC stress and thus results in much shorter lifetime of power managed cores as shown in Figure 6. In both cases reliability is plotted as a function of time in seconds. Our results clearly highlight that power management does not necessary deliver better reliability as it can be heavily influenced by multiple factors, including thermal cycling failure mechanism. As technology scales down, limitations set by TC are going to become an important factor in design for reliability and optimization of power consumption. Thus, the methodology we present in this paper will become an essential part of a the design process.

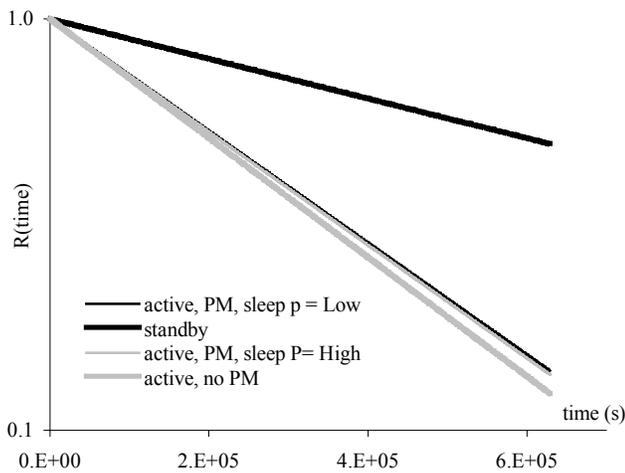


Figure 5. Reliability of a core with larger feature size

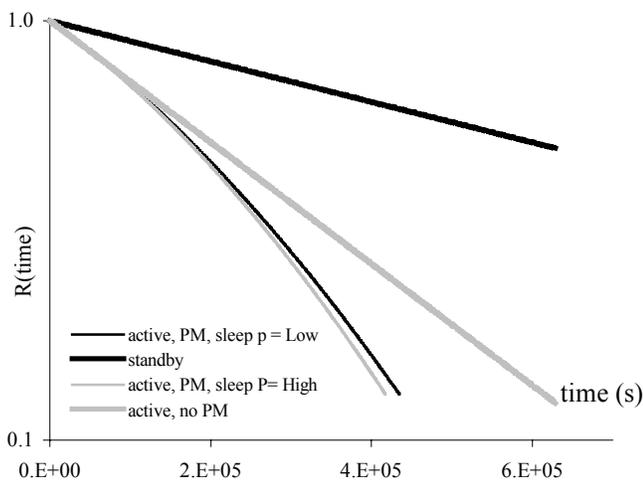


Figure 6. Reliability of a core with smaller feature size

Figure 7 show the dependency of system reliability on redundancy model. Two models are considered, both based on the system shown in Figure 4. One has two cores in active redundancy and the other is with the same cores in standby redundancy. In both cases cores are scheduled to fail at a predefined time. Since all cores have small feature size, frequent switching between power states causes a proportionally larger probability of core failure due to thermal coupling. Clearly it is advantageous to use standby redundancy as it improves both reliability and power consumption. In addition, there is very little difference between standby models with and without power management. Therefore, such models should be considered for a design of an efficient and reliable system. Note that standby mode means that the component is not processing data, and thus it could be in either idle or sleep power state depending on the power management policy. If however, the system needs to be designed with active components potentially due to performance issues, then a special attention needs to be paid to find a PM policy that result in best power

consumption and MTTF. In this particular case, the system with active components that are not power managed takes 65% more energy and fails 14 hours sooner than the system with standby redundancy that uses power management.

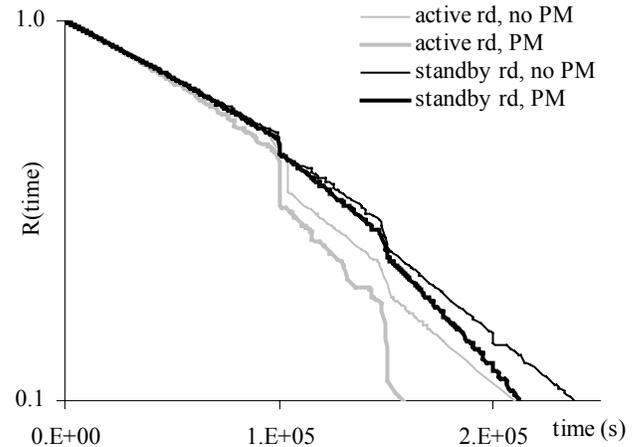


Figure 7. System reliability with redundancy models

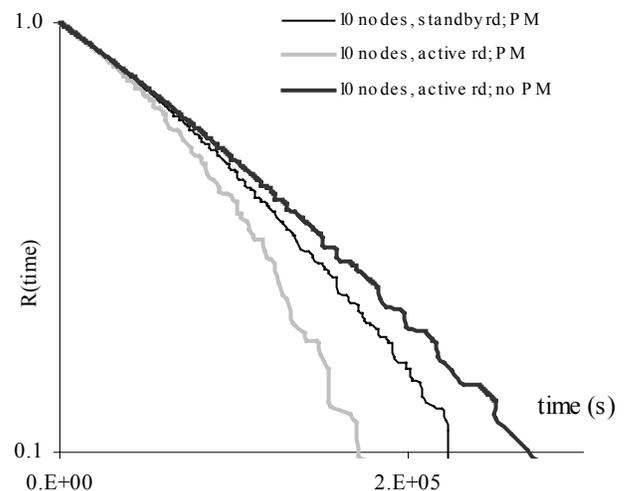


Figure 8. Reliability of a ten core system

Lastly we simulate a more complex system consisting of 10 cores with small feature size (thus frequent transitions to sleep state cause a decrease in system reliability). Five cores are active and the other five are redundant. Figure 8 shows the overall system reliability as a function of time. Three different simulation results are presented, one with no power management, and the other two have power management enabled for both active and standby redundancy. Clearly both the system reliability and the average power consumption are better for power managed cores with standby redundancy than with active redundancy. On the other hand, due to a large negative effect of frequent switching between power states, the best reliability results are obtained with all redundant cores active with no power management, but this also causes a significant increase in average power consumption. The results from this simulation would have been difficult to obtain analytically as the system is sufficiently large and

complex. Our approach at integrating system reliability with power management enables fast and accurate evaluation of the overall system design.

VI. CONCLUSION

In this work we present a first attempt at integrating reliability models with dynamic power management for integrated systems. We determine the system level reliability as a function of failure rates, system configuration and power management policies. Our results show a strong relationship between power management policy and system reliability. System performance remains constant until the system fails due to reliability issues in all cases we consider. In larger feature sizes aggressive power management techniques are helpful for improving system power consumption with little or no cost to system reliability. For smaller feature sizes it becomes critical to carefully trade off the design of power management policy with reliability. Our methodology enables designers of SoCs and NoCs to quickly evaluate their system design in terms of three main objectives: minimum power consumption, maximum system reliability, and optimum performance.

REFERENCES

- [1] E. Y. Wu et al. Interplay of voltage and temperature acceleration of oxide breakdown for ultra-thin gate dioxides. In *Solid-state Electronics Journal*, 2002.
- [2] L. Benini, G. De Micheli, "Networks on Chips: A New SoC Paradigm," *IEEE Computer*, pp. 70-78, Jan. 2002.
- [3] P. Guerrier, A. Greiner, "A Generic Architecture for on-chip packet switched interconnections," *DATE*, pp. 250-256, 2000.
- [4] S. Kumar et al., "A NOC architecture and design methodology," *ISVLSI*, pp. 105-112, 2002.
- [5] E. Rijpkema et al., "Trade-offs in the design of a router with both guaranteed and best-effort services for NOCs," *DATE*, pp. 350-355, 2003.
- [6] A. Jantsch, H. Tenhunen, "Networks on Chip," Kluwer Academic Publishers, 2003.
- [7] International Technology Roadmap for Semiconductors: 2001.
- [8] T. Ye, L. Benini, G. De Micheli, "Analysis of Power Consumption on Switch Fabrics in Network Routers," *Design Automation Conference*, pp. 600-605, 2002.
- [9] M. Wan, H. Zhang, V. George, M. Benes, A. Abnous, V. Prabhu, J. Rabaey, "Design Methodology of a Low-Energy Reconfigurable Single-Chip DPS System," *Journal of VLSI Signal Processing*, 2000.
- [10] M. D. Beaudry. Performance-related reliability measures for computing systems. *IEEE Transactions on Computers*, c-27(6):540-547, June 1978.
- [11] P. Shivakumar et al. Modeling the Effect of Technology Trends on the Soft Error Rate of Combinational Logic. In *International Conference on Dependable Systems and Networks*, 2002.
- [12] P. Shivakumar et al. Exploiting microarchitectural redundancy for defect tolerance. In *21st International Conference on Computer Design*, 2003.
- [13] T. Simunic, S. Boyd, "Managing Power Consumption in Networks on Chips," *Design, Automation and Test in Europe*, pp. 110-116, 2002.
- [14] A. Bavier, A. Montz, L. Peterson, "Predicting MPEG Execution Times," *SIGMETRICS*, pp.131-140, 1998.
- [15] Critical Reliability Challenges for the International Technology roadmap for Semiconductors, International Sematech Technology Transfer document 03024377A-TR, 2003.
- [16] T. Simunic, L. Benini, P. Glynn, G. De Micheli, "Event-driven Power Management," *IEEE Transactions on CAD*, pp.840-857, July 2001
- [17] T. M. Austin. Diva: A reliable substrate for deep submicron microarchitecture design. In *Proc. of the 32nd Annual Intl. Symp. on Microarchitecture*, 1998.
- [18] E. Rotenberg. Ar/smt: A microarchitectural approach to fault tolerance in microprocessors. In *International Symposium on Fault Tolerant Computing*, 1998.
- [19] L. Spainhower and T. A. Gregg. Ibm s/390 parallel enterprise server g5 fault tolerance: A historical perspective. In *IBM Journal of Research and Development*, September/November 1999.
- [20] Jayanth Srinivasan, Sarita V. Adve, Pradip Bose, Jude Rivers, Chao-Kun Hu, "RAMP: A Model for Reliability Aware MicroProcessor Design," *IBM Research Report, RC23048 (W0312-122)* December 29, 2003
- [21] C. J. Hughes, J. Srinivasan, and S. V. Adve. Saving energy with architectural and frequency adaptations for multimedia applications. In *Proc. of the 34th Annual Intl. Symp. on Microarchitecture*, 2001.
- [22] A. Buyuktosunoglu et al. Energy efficient co-adaptive instruction fetch and issue. In *Proc. of the 30th Annual Intl. Symp. on Comp. Architecture*, 2003.
- [23] J. Srinivasan and S. V. Adve. Predictive dynamic thermal management for multimedia applications. In *Proc. of the 2003 Intl Conf. on Supercomputing*, 2003.
- [24] Phillip Stanley-Marbell, Diana Marculescu, "Dynamic Fault-Tolerance Management in Failure-Prone and Battery-Powered Systems," *IWSC ?*
- [25] J. Xu, W. Wolf, J. Henkel, S. Chakradhar, T. Lv, "A case study in NOC design for embedded video," *DATE* 2004.
- [26] H. Jang, M. Kang, M. Lee, K. Chae, K. Lee, K. Shim, "High-level system modeling and architecture exploration with SystemC on a NOC SoC: S3C2510 case study," *DATE* 2004.
- [27] "AMBA Specification," ARM Inc, May 1999.
- [28] "The CoreConnect Bus Architecture," IBM, 1999.
- [29] Jayanth Srinivasan, Pradip Bose, Jude Rivers, "The impact of Technology Scaling on Processor Lifetime Reliability," *UIUC CS Technical Report UIUCDCS-R-2003-2398*, December 2003