

Use It Or Lose It: Wear-out and Lifetime in Future Chip Multiprocessors

Hyungjun Kim[†], Arseniy Vitkovskiy[‡], Paul V. Gratz[†], Vassos Soteriou[‡]

[†] Department of Electrical and Computer Engineering, Texas A&M University

[‡] Department of Electrical Engineering, Computer Engineering and Informatics, Cyprus University of Technology

hyungjun@tamu.edu, arseniy.vitkovskiy@cut.ac.cy, pgratz@tamu.edu, vassos.soteriou@cut.ac.cy

ABSTRACT

Moore's Law scaling is continuing to yield even higher transistor density with each succeeding process generation, leading to today's multi-core Chip Multi-Processors (CMPs) with tens or even hundreds of interconnected cores or tiles. Unfortunately, deep sub-micron CMOS process technology is marred by increasing susceptibility to wearout. Prolonged operational stress gives rise to accelerated wearout and failure, due to several physical failure mechanisms, including Hot Carrier Injection (HCI) and Negative Bias Temperature Instability (NBTI). Each failure mechanism correlates with different usage-based stresses, all of which can eventually generate permanent faults. While the wearout of an individual core in many-core CMPs may not necessarily be catastrophic for the system, a single fault in the inter-processor Network-on-Chip (NoC) fabric could render the entire chip useless, as it could lead to protocol-level deadlocks, or even partition away vital components such as the memory controller or other critical I/O. In this paper, we develop critical path models for HCI- and NBTI-induced wear due to the actual stresses caused by real workloads, applied onto the interconnect microarchitecture. A key finding from this modeling being that, counter to prevailing wisdom, wearout in the CMP on-chip interconnect is correlated with lack of load observed in the NoC routers, rather than high load. We then develop a novel wearout-decelerating scheme in which routers under low load have their wearout-sensitive components exercised, without significantly impacting cycle time, pipeline depth, area or power consumption of the overall router. We subsequently show that the proposed design yields a $13.8\times$ - $65\times$ increase in CMP lifetime.

Categories and Subject Descriptors

B.4.3 [INPUT/OUTPUT AND DATA COMMUNICATIONS]: Interconnections (Subsystems); B.8.1 [PERFORMANCE AND RELIABILITY]: Reliability, Testing, and Fault-Tolerance

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

MICRO'13 December 7-11, 2013, Davis, CA, USA.

Copyright ©2013 ACM 978-1-4503-2638-4/13/12 ...\$15.00.

General Terms

Reliability, Design

Keywords

Negative Bias Temperature Instability (NBTI), Hot Carrier Injection (HCI), Wearout, Network-on-Chip, Reliability, Lifetime

1. INTRODUCTION

The continuous aggressive miniaturization of CMOS feature sizes and the resulting increase in transistor density has recently sparked the multicore era. Architects have harnessed this increasing supply of transistors, resulting in the design of parallel systems, including Chip Multi-Processors (CMPs) [15]. In these systems, the on-chip interconnect, typically organized as a Network-on-Chip (NoC) [10], plays a vital role in enabling communication among the various on-chip computational, memory and peripheral components, as illustrated in Figure 1. Unfortunately, deep sub-micron CMOS process technology is marred by increasing susceptibility to wearout, dramatically shortening the useful lifespan of such on-chip parallel systems. Recent ITRS reports indicate a 10-fold decrease in wear-rate will be required to maintain current design lifetimes without dramatically increasing timing margins [17]. As we will illustrate, wearout does not affect all components equally; wear of the cores can often be managed, while wear of the NoC interconnect can be catastrophic. Furthermore, we will show that wear in the NoC is highly dependent upon the actual operational stresses caused by real CMP workloads. In this work, we develop techniques to *proactively* maintain the CMP NoC in the face of workload-dependent wear, and hence improve the overall functional lifetime of the CMP as a whole.

Two key operational stress-induced wear mechanisms in current and future CMOS technology are Hot Carrier Injection (HCI) and Negative Bias Temperature Instability (NBTI) [27]. Both HCI and NBTI lead to a shift of the transistor's threshold voltage, eventually leading to switching delay and critical path degradation [18]. Though these effects do not result in circuit opens or shorts, over time they can cause critical path timing violations. Given equivalent supply voltage and temperature, HCI and NBTI degradation are primarily dependent upon the time transistors have been operating under stress. These types of stresses are primarily data- and usage-dependent, in terms of the activity factor (i.e., the fraction of cycles in which a transistor switches) and duty cycle (i.e., the percentage of time the gate's volt-

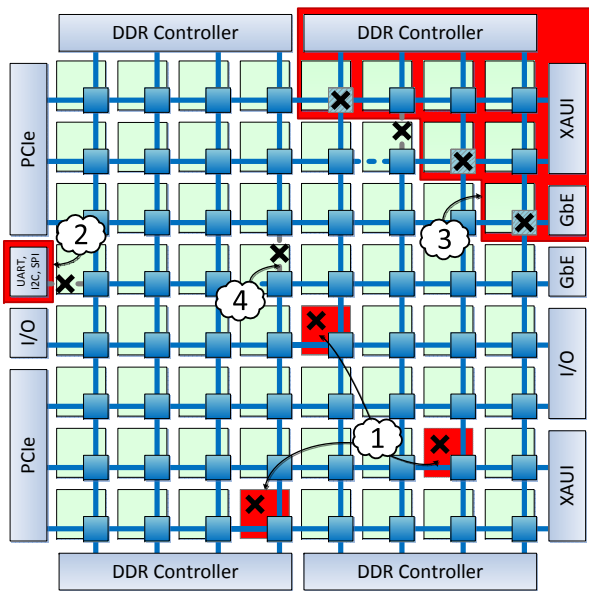


Figure 1: A 64-core CMP interconnect with an 8×8 2D mesh NoC. Components marked with a black \times illustrate wearout failure. The failure scenarios are as follows: (1) failure of cores; (2) peripheral device disconnected from the system due to link failure; (3) network segmentation resulting in a disconnected sub-network; (4) individual link failure.

age is held at a constant zero), respectively, of the gates in typical CMOS logic circuits.

Figure 1 illustrates a CMP exposed to wearout failures in various of its components. As prior work would indicate, individual core wearout and failure need not be catastrophic to the functionality of many-core CMPs due to the inherent core redundancy that a CMP implies [19, 9, 34, 29, 23, 16]. With increasing numbers of cores, a proportionally smaller portion of the overall system’s required throughput is dependent upon each individual core. The component failure scenario (1) of Figure 1 shows this case. Failure caused by wearout of some cores need not result in full-system failure. Instead the system could suffer some performance loss while preserving correct functionality, assuming core-level error detection and appropriate system support is available [9, 34, 29, 23, 16].

For the NoC interconnecting the cores, however, the assumption of redundancy-based wear resilience breaks down, (c.f., component failure scenarios (2), (3) and (4) of Figure 1). Scenario (2) illustrates the case where a wearout-induced link failure precludes access to a key I/O peripheral, while in scenario (3) link and router wearout has partitioned away a large fraction of the network, making those cores and I/O components inaccessible from the rest of the system. In both cases, wearout is catastrophic, in that the system will likely be rendered unusable due to these failures, unlike the core wearout in scenario (1) discussed earlier. Even scenario (4), in which a single link is broken due to wear-induced failure, might lead to a communication protocol-induced deadlock(s), or subnetwork isolation, if the network is not provisioned to address wear-induced failures.

Prior work has proposed various fault-tolerant routing algorithms and fault-insensitive router and link designs in an

attempt to manage faults as they occur [38, 32, 12, 6, 5, 11], however, network isolation and key resource partitioning cannot be fully resolved using only such *reactive* techniques. Ideally, one would prefer to develop *proactive* mechanisms to extend the healthy status of the system without failure, rather than react to the faults once they occur. Such proactive mechanisms could be coupled to the reactive mechanisms, in the hopes that the latter would be required less frequently as faults in the system would occur less frequently.

In this work we present such a proactive technique, designed to decelerate the effects of aging in the NoC of a CMP. Based upon detailed HCI and NBTI transistor-level aging models, we develop a novel, critical path-based model to characterize the effects of aging-related wear. Based upon this model we analyze the NoC router microarchitecture to find the paths most susceptible to wearout. Using real workloads from the PARSEC benchmark suite [7], we characterize various wearout mechanisms that map onto those paths. Finally, we develop a wearout-resistant router microarchitecture which prolongs circuit lifetime through targeted mitigation techniques with negligible influence on the router’s timing, pipeline, CMOS area requirements, and power consumption. This proposed technique yields a $13.8 \times 65 \times$ increase in CMP lifetime.

This paper provides the following contributions:

1. Generalized, microarchitecture-level (rather than device-level) HCI and NBTI wearout-induction models.
2. Characterization of NoC router and link wearout due to HCI and NBTI under real workloads from the PARSEC benchmark suite [7].
3. A novel wear-resistant router microarchitecture which dramatically improves interconnect lifetime, and hence full-system survivability in the presence of both HCI- and NBTI-wearout mechanisms.

This paper is organized as follows. Section 2 examines existing transistor-level models for HCI- and NBTI-induced wear. Next, Section 3 examines the sensitivity of the router’s critical path to wear by analyzing the activity and duty cycle of its critical path, and characterizes the router’s wear caused by the behavior of real application workloads. Based upon these wearout models, Section 4 then develops a circuit path delay model for workload stress-induced wear. Section 5 proposes a novel router microarchitecture to improve the lifetime of NoC routers under realistic workloads, while Section 6 evaluates the proposed design. Finally Section 7 presents prior related work, while Section 8 concludes.

2. BACKGROUND

Prior research shows that the two dominant CMOS transistor physical failure mechanisms are Hot-Carrier Injection (HCI) and Negative Bias Temperature Instability (NBTI) [26]. Under both failure mechanisms charge becomes trapped in or near the gate oxide resulting in a slow increase of the transistor threshold voltage (V_{th}). This in turn causes the delay in transistor state switching to expand.

In traditional synchronous circuit CMOS designs, the clock frequency of a given design is determined by the circuit path which exhibits the longest latency between its end latches, within a given system design. This *critical path* comprises a chain of connected gates between latches. As HCI- and NBTI-induced aging progresses, it gradually extends the delay of each gate found in this chain, slowing down the en-

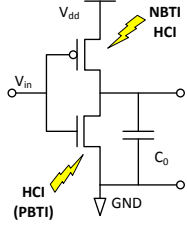


Figure 2: CMOS inverter.

tire critical path. In modern CMOS designs, due to this age-induced slow-down, and other causes, such as process variation [20], designs are given timing guard-bands so as to guarantee their intended functionality for a certain duration of time [2]. Once the aggregate increase in delay along a timing-critical path exceeds this guard-band, due to the aggregation of increasing delays occurring in individual gates along this path, the functionality of the system is no longer assured. The moment at which this timing violation first occurs determines the system’s useful life span. Of course, HCI and NBTI impact all transistors in the design (not only those in the critical path), however, those on the critical path are more likely to exceed the guard-band causing a critical failure.

In this section, we first describe the impact of these aging mechanisms upon V_{th} using transistor-level analytical models. We then examine a number of specific NoC router critical paths which are most susceptible to these aging effects, since they determine the system’s clock rate.

2.1 Failure Mechanisms

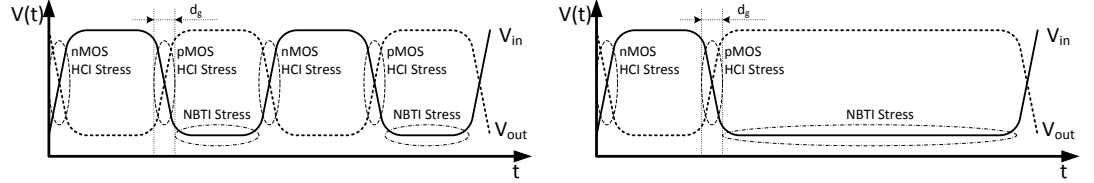
Design rules and operating conditions are precisely chosen to ensure correct product functional operation over its intended lifetime [22]. To obtain a given level of performance, when utilizing an integrated circuit, under various design constraints, it becomes imperative to create and analyze the reliability model of the digital system under consideration and design.

As previously discussed, the HCI and the NBTI mechanisms do not induce failures, rather they shift parameters over time under circuit operational stresses. The Reaction-Diffusion (R-D) model uses the threshold voltage (V_{th}) shift as a proxy of NBTI and HCI stress [37]. The V_{th} shift causes transistor delay degradation according to the Alpha Power Law [31]:

$$d_g \propto \frac{V_{dd}}{\mu(V_{dd} - V_{th})^\alpha} \quad (1)$$

where d_g is the transition delay, $\mu \propto T^{-1.5}$ (T being Temperature) and $\alpha = 1.3$.

Lifetime can be defined as the time until an important material of a component or device parameter degrades beyond the point at which the device or circuit can function properly in its originally intended application. For a single gate, when ΔV_{th} reaches some level (in practice, it is usually 10% [37]) the transistor is considered to be over-aged. For the multi-gate path, the cumulative transistor delay shift increases faster than a single gate’s worst-case delay degradation. Therefore, a total gate delay shift over the entire



(a) Higher switching activity, lower duty cycle. (b) Lower switching activity, higher duty cycle.

Figure 3: HCI and NBTI stress time windows for a CMOS inverter.

path, when its value reaches or exceeds the 10% threshold mark, can serve as a lifetime period indicator.

Figure 2 shows a typical CMOS inverter, indicating the failure mechanisms associated with each type of transistor: HCI affects both the nMOSFET and pMOSFET transistors, while NBTI affects only the pMOSFET transistor (note that PBTI is the complement of NBTI and affects nMOSFET transistors only, however, its effect is generally considered to be much smaller than that of NBTI). The following subsections present a device parameter degradation model that captures the HCI and NBTI wearout effects.

2.1.1 Hot Carrier Injection (HCI):

Hot Carrier Injection (HCI) is a wear-out mechanism which occurs when carriers flow along the channel in MOSFET transistors and gain sufficient kinetic energy to be injected into the gate oxide resulting in a charge trap and interface state generation. This leads to a gradual transistor parameter shifting, including switching frequency degradation, rather than causing an immediate failure event [18].

A substrate current-based (I_{sub}) model is commonly used to estimate the effect of HCI. Prior work shows that the transistor threshold voltage shift due to HCI under DC stress is

$$\Delta V_{th_HCI}|_{DC} = A(I_{sub})^m t_{stress}^{n'}, \quad (2)$$

where A is the material-dependent parameter, t_{stress} is the stress time, and n' and m are technology-related exponents [22, 18].

According to the Alpha Power Law (1), the delay of a transistor depends linearly on threshold voltage for small shifts, so the gate delay shift can be expressed as

$$\Delta d_{g_HCI}|_{DC} = \hat{A}(I_{sub})^m t_{stress}^{n'}, \quad (3)$$

where \hat{A} is a fitting constant.

The lifetime of a device exposed by a direct HCI effect is [18]

$$TTF_{HCI}|_{DC} = A_{HCI} (I_{sub})^{-N'} e^{\left(\frac{E_{g_HCI}}{kT}\right)}, \quad (4)$$

where E_{HCI} is the apparent activation energy, I_{sub} is the substrate current under stress at $V_G = V_D$, T is the runtime temperature, k is the Boltzmann’s constant, N' is the technology-related exponent, and A_{HCI} is a fitting constant.

HCI stresses the device only during dynamic transitions when current flows through the device. Figure 3 shows voltage waveforms of a standard CMOS inverter. The pMOSFET transistor suffers HCI stress when the output of the inverter is pulling-up and C_0 is charging up (see Figure 2).

The nMOSFET transistor experiences HCI degradation during the reverse dynamic stage, when the output of the inverter is discharged to the ground (low-voltage level) [22]. Thus, each of the CMOS transistors experiences degradation only during half of a switching period, and hence the relation between HCI stress time t_{stress}^{HCI} and run-time t can be derived as

$$t_{HCI_stress} = d_g f \alpha_{SA} t, \quad (5)$$

where d_g is the transition delay, α_{SA} is the switching activity, and f is the clock frequency.

Since HCI stress occurs during the device turn-on and turn-off periods, the impact of HCI under AC stress can be extracted from equations (3) and (4) using (5)

$$\Delta d_{g_HCI}|_{AC} = A(I_{sub})^m (d_g f \alpha_{SA} t)^{n'}, \quad (6)$$

And finally, the last equation is transformed into a relation for HCI lifetime

$$TTF_{HCI}(T, \alpha_{SA})|_{AC} = A_{HCI} \frac{1}{d_g f \alpha_{SA}} (I_{sub})^{-N'} e^{\left(\frac{E_a_{HCI}}{kT}\right)}. \quad (7)$$

This relation shows that the lifetime of a transistor due to HCI is inversely related to the switching activity α_{SA} of the gate input. Hence, frequent switching, such as shown in Figure 3a, not only increase the dynamic power consumption, but also speed-up the aging effect, whereas a gate with less frequently occurring transitions, as shown in Figure 3b, will experience lighter HCI-induced aging.

2.1.2 Negative Bias Temperature Instability (NBTI):

Negative Bias Temperature Instability (NBTI) is a wear-out effect that influences pMOSFET transistors as long as they operate in inversion (i.e., a “0” voltage on the input of an inverter, as shown in Figure 2). Thus the data-dependent stress caused by NBTI is very different from that of HCI, which is under stress during voltage level switching. NBTI changes the pMOSFET transistor parameters over time. In particular, it leads to an increase in the threshold voltage (V_{th}), as well as a reduction in the drive current due to charge carrier mobility degradation. As with HCI, NBTI does not result in complete circuit failure, but rather in circuit speed degradation. JEDEC reports that process technology scaling will lead to a larger NBTI-induced threshold voltage in pMOSFET transistors [18]. It has been reported that, unlike HCI, some degree of recovery from NBTI degradation can occur in the event that a relaxation period occurs after the stress period [18, 22, 37].

We use the AC stress model for NBTI degradation under high-frequency CMOS operation, as proposed by Lu et al. [24], which provides a theoretical upper bound estimation of the NBTI effect in terms of time as

$$\Delta V_{th_NBTI} = A \left(\frac{\beta}{1-\beta} \right)^n t^n e^{\left(-\frac{nE_a_{NBTI}}{kT}\right)}, \quad (8)$$

where $E_{a_{NBTI}}$ is the apparent activation energy, T is the run-time temperature, t is the operating time, k is Boltzmann’s constant, n is the time exponent, and A is a fitting constant [24].

According to the Alpha Power Law (1), the first-order gate delay can be approximated as a linear function of the threshold voltage. Hence, the gate delay shift can be ex-

pressed as

$$\Delta d_{g_NBTI} = \hat{A} \left(\frac{\beta}{1-\beta} \right)^n t^n e^{\left(-\frac{nE_a_{NBTI}}{kT}\right)}. \quad (9)$$

The lifetime of a single transistor under AC stress can be derived from (9) as

$$TTF_{NBTI} = \left[A_{NBTI} \left(\frac{1-\beta}{\beta} \right)^n e^{\left(\frac{nE_a_{NBTI}}{kT}\right)} \right]^{1/n}. \quad (10)$$

Thus, lifetime degradation due to NBTI depends on the duty cycle of the input signal. Transistors which with a smaller duty cycle, such as the duty cycle shown in Figure 3a in comparison to the duty cycle shown in Figure 3b, experience a slower degradation rate.

2.1.3 HCI and NBTI failure mechanism analysis:

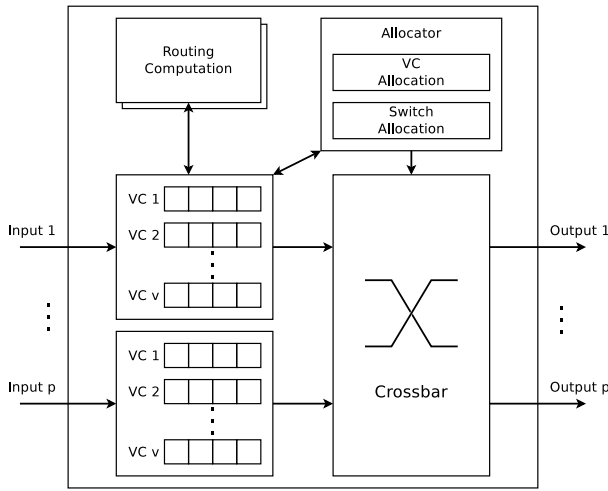
It may first appear that a technique which improves device lifetime by decreasing NBTI must come at the cost of a comparable degradation caused by HCI-related wear (and vice-versa). We note, however, that the activity factor α_{SA} is not the inverse of duty cycle β ; when β is large, it is possible to make a substantial change to β without proportionally impacting α_{SA} . Furthermore, because of the $\frac{1}{(1-\beta)}$ term in Equations 8 and 9, large β s tend to have a disproportionate impact on aging-related slow-down. Even a small improvement in the value of β can therefore have a substantial positive effect on the overall device lifetime (especially when β is relatively large).

2.2 Router Microarchitecture

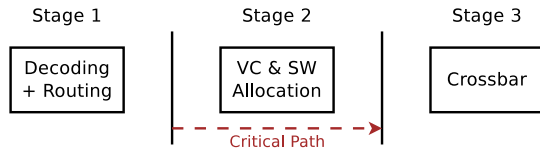
The canonical NoC virtual channel router was proposed by Peh and Dally [28]. Its block diagram is shown in Figure 4a. The major building blocks of this NoC router are input channels, a crossbar (switch), and the control logic which includes the switch and virtual channel allocators. When used in a 2-D Mesh NoC architecture, typically five input and output channels, p , are used to connect its four immediate neighbors at the cardinal points, and its local processing element. An input channel is composed of a given number of virtual channels (VCs), each of which includes registers to keep track of their statuses, and buffers to store flits (flow-control units, a logical fixed-segment of a packet). The routing units also examine flits found in the input channels to determine the next-hop direction packets should take (i.e., the east, west, north or south directions). The VC allocator assigns a free VC at a downstream router to a head flit, the first flit of a packet. If the head flit successfully obtains a VC, it competes with any other flits destined to the same output port during switch allocation. Body and tail flits in the same packet skip the routing and VC allocation stages, and directly proceed to the switch allocation stage. Once switch allocation is complete, the flit traverses the crossbar.

Our baseline router performs both VC and switch allocation during the same cycle by speculatively allowing a packet to compete for the switch while it is still competing for a free VC at the downstream router [28]. Figure 4b shows the baseline router pipeline. Flit decoding and routing computation are done in *Stage 1*. The combined VC and switch (SW) allocations are done in *Stage 2*. In *Stage 3*, flits traverse the crossbar.

NBTI and HCI both slow transistor switching, thus, the damage from the circuit aging first takes place where timing



(a) Router Block Diagram



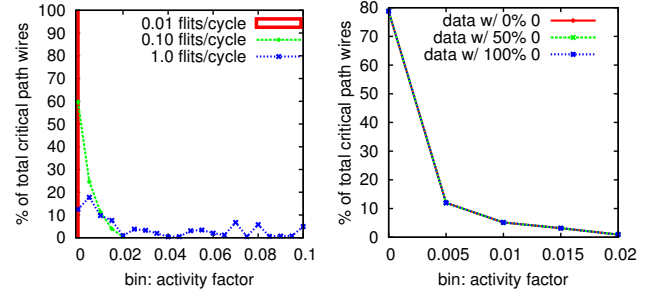
(b) Router Pipeline Stages

Figure 4: Baseline Router

is most critical. To determine the critical path of the baseline router, we adapted our baseline RTL from the publicly available router RTL model designed by Becker [4]. This RTL model was synthesized using Synopsys Design Compiler and mapped to TSMC 45 nm standard cell library to a 1 GHz frequency. All critical paths with slack less than 10% of the clock frequency were gathered using Synopsys Design Vision and analyzed off-line.

The results of this analysis are highlighted in Figure 4b. We found that all timing critical paths (i.e., those within 10% of the 1 GHz clock frequency), pass through the VC and switch (SW) allocators. These results correspond well with prior research [28]¹. The utilization of the allocators is closely related to the router’s incoming rate, or the number of flits traversing the given router per cycle, because the allocators are enabled by the input channel which sends the request signal to the allocators when it has flits to forward. As the critical path is initiated by the request signal, the wire activity along the path is dependent upon how often the request signal is set, which in turn is determined by the workload’s utilization of that router. We therefore expect that the stress time for HCI and NBTI, which are closely related to the activity factor and the duty cycle, thereby should be also closely correlated to the router’s utilization. In the next section, we perform an in-depth study on the impact of typical CMP workloads on the router’s critical path in terms of activity factor and duty cycle.

¹Note: Some prior work highlights the credit return path as a critical path within the router; in our experiments assuming 6 mm links between routers we found that the credit return path was not on the critical path. With longer links, however, the return path might become critical requiring further analysis.



(a) Activity factor with respect to router incoming rate. (b) Activity factor with respect to data content (the packet’s data bits percentage at zero).

Figure 5: Sensitivity to the activity factor.

3. ROUTER WEAR CHARACTERIZATION

To examine the sensitivity of the router’s critical path to HCI and NBTI wear, we first analyze the activity of wires residing in the baseline router under synthetic workloads. The router has 5 physical channels, 4 VCs per physical channel, and 4 flit-deep buffers per VC. Dimension-order routing (DOR) is used. The network is designed to transfer 64-byte memory blocks where the link-width between adjacent routers is 128 bits, discounting any flow-control signals. Hence, if a packet includes such data it is composed of 5 flits (1 head flit containing routing information and meta-data, and 4 data flits), otherwise, it is composed of only 1 head flit. The workload is generated with an arbitrary injection rate, maintaining a 50-50% proportion of 1-flit and 5-flit packets. As described in Section 2.2, all paths from the post-synthesis router model with 10% or less slack relative to the 1 GHz clock frequency, were examined. In particular, information about the activity factor and duty cycle of all wire nodes along each critical path under these workloads was retained for analysis.

3.1 Impact of Workload upon Router Activity Factor

HCI is proportional to the activity factor of the interconnect NoC wires, such that a higher activity factor results to an accelerated (higher) aging rate (refer to Section 2.1.1). Figure 5a shows the histogram of activity factors of the wires on the critical paths of an NoC router with respect to varying incoming rates. The first observation we make is that the nodes have a quite low activity factor, the vast majority switching less than 10% of the time (activity factor of 0.1).

Intuitively, the higher incoming rate should cause a correspondingly higher activity factor. This implies that a router experiencing traffic from an application that injects more frequently ages at a faster rate. Hence, it is desirable to reduce (or keep low) the incoming rate so as to improve the longevity of the router. We find, however, the activity factor does not appear to be very sensitive to the incoming rate, in that it does not increase as much as the incoming rate increases. For instance, even at the very high incoming rate of 1.0 flits/cycle, the activity factors of most of the wires remain at a relatively low value and only 7.7% of wires have an activity factor greater than 0.1.

Without a priori knowledge of the router’s critical path,

one might expect that the content of the data traversing the network would also affect the activity of routers. Figure 5b shows the histogram of activity factor with various data contents (percentage of logical zeros in each data-flit vector) at a fixed incoming rate of 0.10 flits/cycle, to examine the router’s critical path activity factor sensitivity to data content. As expected, the data content does not affect the activity factor of the wires along the router’s critical path.

3.2 Impact of Workload upon Router Duty Cycle

NBTI is highly sensitive to the duty cycle of gates (see Section 2.1.2). Figure 6a depicts the histogram of wires along the critical path at a given duty cycle for different incoming rates, in terms of flits per cycle. In the figure, the width of each bin is 0.05; hence bin[0] shows the percentage of gates with duty cycle in the range of [0, 0.05) and so on. In general, the majority of gates fall into the duty cycle bins near 0, 0.5 or near 1.0, regardless of the incoming rate. As the incoming rate grows, the bins at the two ends of the spectrum fall while the central part moves up, indicating that increasing flit incoming rate causes less skew in the duty cycle towards these extremes. Figures 6b and 6c magnify the two ends of Figure 6a. To improve observability in these figures we use a narrower bin width of 0.001. With the increased resolution we note that higher incoming rates have a great impact on duty cycle at these extremes, reducing the percentage of wires along the critical path with the highest and lowest duty cycle from ~35% to near 0%.

The incoming rate’s effect on duty cycle causes notable differences in NBTI’s impact upon gate delay. As shown in Equation (9), there is a non-linear relationship between duty cycle and gate delay such that the gate delay shoots up as the duty cycle gets closer to 1.0. Hence, for example, having two gates in a given path with the same duty cycle of 0.5, is better than having two gates with duty cycle of 0.0 and 1.0, respectively, in terms of the path-cumulative impact of NBTI upon gate delay. The delay increase due to NBTI from a single gate under a duty cycle of 1.0 alone will greatly exceed the sum of delay increases from two gates, each with a duty cycle of 0.5. Thus, it is preferable to have a higher incoming rate, with more gates with duty cycles closer to 0.5, than to have a lower incoming rate and gates that have duty cycles closer to 1.0; although it is notably counterintuitive that accelerated wear-out occurs when routers are under-utilized. Based upon these observations we will next develop a multi-gate delay model in Section 4.

We examined the router’s critical paths to determine why these paths exhibited such a skewed duty cycle and low activity factor (see Section 3.1). In the router, the longest paths through the crossbar and VC allocators are primarily concerned allocation corner-cases, such as multiple simultaneous incoming packets attempting to allocate a VC with limited available VCs. These cases are relatively rare, only occurring under highly loaded network conditions, thus these control signals switch infrequently and have very poor duty cycles when the NoC experiences low loads.

3.3 Workload Characterization

Having identified the per-router incoming rate to be a critical workload characteristic that correlates with wear, we now examine the router-to-router incoming rate vari-

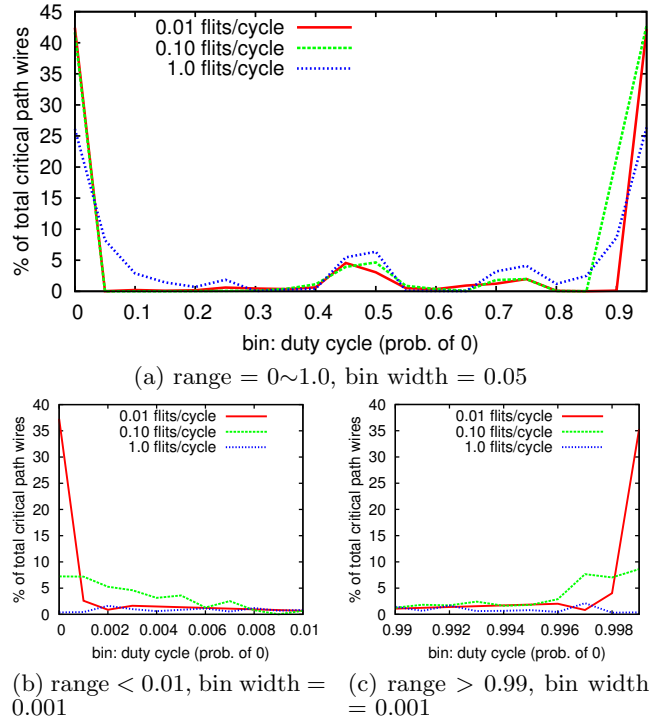


Figure 6: Histogram of duty cycle w.r.t incoming rate.

ance in realistic workloads. In this study, we use the PARSEC benchmark suite as our workload, as these benchmarks mimic a range of representative next-generation large shared-memory multi-threaded programs for CMPs [7]. The diversity of the PARSEC benchmarks makes them especially useful for this study, as they span a diverse range of emerging applications with varying on-chip communication spatio-temporal characteristics. Specifically, with the PARSEC benchmarks one observes different and varying behaviors in the NoC’s packet (or flit) incoming rate, as will be outlined next in detail.

The realistic workloads are generated from the gem5 simulator [8] emulating a 64-core system executing multithreaded programs in the PARSEC v2.1 [7] suite. Table 1 shows the details of the system setup used in our simulations. We first generate NoC traffic for each application for its Region of Interest (ROI). We then count the number of flits traversing each router, to compute incoming rate of that router. We note that the term *incoming rate* here is the number of flits injected to a particular router, per unit time, rather than the number of flits generated and injected to the network as a whole. This includes the number of flits generated by the router’s local processing element and the flits going through or headed for the router. The reason for concentrating on the incoming rate temporal characteristics is that the router’s critical path activity is highly related to the frequency of flit arrival (see Sections 3.1 and 3.2).

Figure 7 depicts the average number of flits injected into a router, shown by the solid bars, according to the aforementioned experimental setup, per unit time for each PARSEC benchmark. The incoming rate at routers, on average, is 0.02 flits per cycle. It varies across the programs under examination ranging from 0.003 (*x264*) to 0.05 (*cannal*). The average incoming rate also varies within the same applica-

Cores	64 on-chip, in-order, Alpha ISA
L1 Cache	32 KB instruction/32 KB data, 4-way, 64 B lines, 3 cycle access time MESI cache coherent protocol
L2 Cache	64 bank fully shared S-NUCA, 16 MB, 64 B lines, 8 way associative, 8 cycle bank access time
Memory	150 cycle access time, 8 on-chip memory controllers
Network	8 × 8 Mesh, X-Y routing, 4 VCs/port, packet length: 1 flit or 5 flits

Table 1: System Setup.

tion based on the cartesian location of the router, and the variance is captured by the dark line over each bar. The bottom of the line shows the incoming rate of the router which handles the least traffic among the routers in the network for that benchmark (*min incoming rate*), and the top of the line denotes the incoming rate of the busiest router (*max incoming rate*). Hence, the per-router incoming rate under the PARSEC workloads actually varies between 0.0005 (*min* of *x264*) and 0.085 (*max* of *canneal*). In Figure 7, “AVG” denotes the arithmetic means of the average incoming rate (the bar) and the *min* and *max* incoming rates (the line) across the entire range of benchmarks. “ALL” captures those three incoming rates when the system runs all the benchmarks, one at a time, sequentially.

In general, the average incoming rate seen at each router is quite low, at 0.02 flits per cycle. Hence, *HCI-induced aging is not expected to contribute significantly to gate delay under these workloads*. Alternately, as discussed previously, a low incoming rate causes NBTI-induced aging. Thus, *routers running PARSEC workloads are exposed to accelerated NBTI-induced aging due to their light traffic*. Furthermore, there is an observable high variance in their spatially distributed network flit incoming rates, such that some routers executing the *ferret* and *x264* benchmarks experience even less than a 0.001 flits per cycle incoming rate. We therefore, focus on NBTI aging in the remainder of this study.

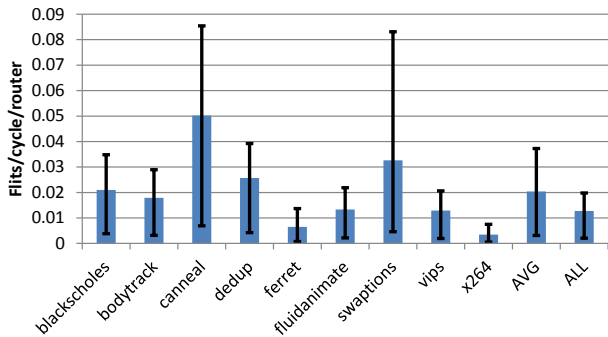


Figure 7: Average flit incoming rate per router for the entire range of the PARSEC benchmark suite (bars) and related range of incoming values (lines).

4. PATH DELAY

In Section 2.1, we examined existing formulas characterizing wear-induced transistor gate delay. While these equations accurately model the incremental breakdown of indi-

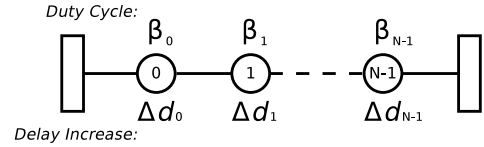


Figure 8: An example circuit path with multiple gates.

vidual gates, we observe that a single gate is only one component of a particular critical path. Here, we derive formulas to compare the relative lifetime of two systems that operate under the same conditions, focusing on the microarchitecture level. There are a number of delay models which take into consideration the aging effect at the transistor level or gate level, but few exist at the microarchitectural-level. Ultimately, to calculate the point at which gate delay compromises timing along a particular path, one must examine the cumulative increase in delay (delta-delay) along that path. Here, we propose a method to compute the relative lifetime of a path between latches, given the duty cycle of each gate along that path.

We assume that a number of sequential gates comprise a circuit path as shown in Figure 8. Along this path, the delay increase due to the *i*-th gate with duty cycle β_i at time *t* can be expressed as:

$$\Delta d_i(\beta_i, t) = \psi \times t^n \times \left(\frac{\beta_i}{1 - \beta_i} \right)^n \quad (11)$$

where the constant ψ includes all other terms in Equation (9) under the assumption that those will remain constant under the same condition. The sum of delay increase along a path with *N* gates at time *t* can be computed as:

$$\Delta d(t) = \sum_{i=0}^{N-1} \Delta d_i(\beta_i, t) = \psi \times t^n \times \sum_{i=0}^{N-1} \left(\frac{\beta_i}{1 - \beta_i} \right)^n \quad (12)$$

A system is reliable as long as the $\Delta d(t)$ of the critical path is smaller than the guard-band. Hence, we define lifetime, $T_{lifetime}$, such that $\Delta d(T_{lifetime}) < guardband$. The Acceleration Factor (*AF*) is defined as the ratio of the lifetime of the system under consideration, $T_{lifetime}(x)$, and a reference system, $T_{lifetime}(ref)$:

$$AF(x) = \frac{T_{lifetime}(x)}{T_{lifetime}(ref)} = \left(\frac{\sum_{j=0}^{M-1} \left(\frac{\beta_j}{1 - \beta_j} \right)^n}{\sum_{i=0}^{N-1} \left(\frac{\beta_i}{1 - \beta_i} \right)^n} \right)^{1/n} \quad (13)$$

where β_i is the duty cycle of the *i*-th gate on the critical path of the system under consideration, and β_j is the duty cycle of the *j*-th gate on the critical path of the reference system. In Equation 13, it is assumed that the number of gates on the critical path of the two systems are *N* and *M*, respectively.

5. LIFETIME-EXTENDING ROUTER MICRO-ARCHITECTURE

As Section 3 outlined, the aging process is incoming rate-dependent along the critical path. The gate delay increases and the timing constraints are violated along the critical path first. A low incoming rate causes a biased duty cycle

in the wires along the critical paths, because those paths deal with allocation corner-cases which are rare unless the load is very high. These biases accelerate NBTI, thus the router requires an increased incoming rate to improve its longevity. However, increasing the incoming rate artificially yields other problems such as increased power consumption and acceleration of the HCI effect. The duty cycle must therefore be improved without increasing the activity factor significantly. We note that, although duty cycle and activity factor are related, it is possible to reduce duty cycle of a node substantially without increasing the activity factor substantially, by *infrequently* changing the value of that node. Hence we propose a method to exercise the critical path, which improves the duty cycle while minimally disturbing activity factor, improving NBTI without substantially impacting HCI. The goals of the proposed mechanism are: (1) to improve the duty cycle by allowing the circuits to operate at a greater portion of their time in the “1” state, without affecting the actual data values being transferred, (2) to not change the state of the router, (3) to not worsen the critical path timing, and (4) to not significantly increase the activity factor.

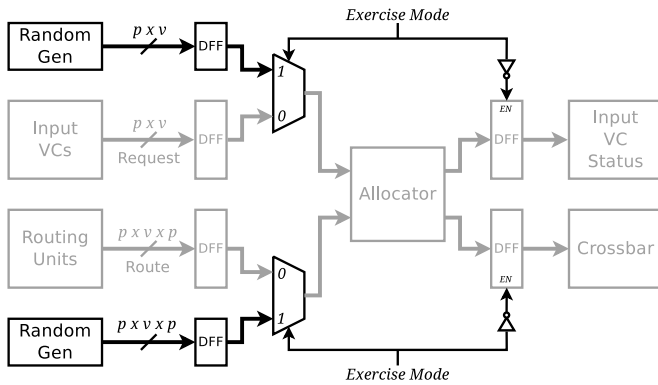


Figure 9: Exercise logic (original hardware in gray, proposed additional exercise logic in black).

Figure 9 illustrates the critical path of a baseline router, along with our proposed modifications to reduce the wear-out effects of NBTI. The gates and wires in black are our additions to the baseline router. The critical paths in an NoC router, as stated in Section 2.2, lie within the Virtual Channel (VC) and crossbar allocation stages handled by the router allocator. Each VC sends a one bit “Request” signal to the allocator to reserve a VC at the downstream router, and/or to bid for switch bandwidth at the crossbar so that the crossbar can be traversed by competing flits. There are p physical channels each of which has v virtual channels, hence, there are $p \times v$ of such control bits in total. Each “Request” signal must be sent with a p bit-width “Route” signal giving the allocator the information as to where the corresponding flit is destined (i.e., to which VC at a physical port downstream). Hence, there are $p \times v \times p$ bits for the “Route” signal. Given the baseline NoC router described in Table 1, this yields a total of 20 bits for Request and 100 bits for Route, or a total of 120 bits of critical path inputs.

We propose to balance the duty cycles of the critical paths within the router through the allocators by injecting random data. Figure 9 depicts how the proposed “exercise logic” works for the “Request” signal. First, the “exercise mode” is

activated when there is no request from the input channels to the allocator. This replaces the “Request” and “Route” signals with a random vector which is generated by a random number generator (“Random Gen”) in the same figure. This random vector alters the duty cycle of wires along the critical path within the allocators by simulating the various allocation corner-cases which rarely occur in typical operation under realistic loads. This allows the duty cycle to approach the theoretically optimal value.

The proposed “exercise mode” should not affect the router functionality. Hence, it is enabled only when no packets reside in the input channels. Also, the random output from the allocator should not propagate to the next pipeline stages, such as the crossbar traversal stage. The states of the input virtual channels, similarly, should not be updated with the false output produced. Thus, the flip-flops (DFF’s in Figure 9) or latches between the allocator and the next stage, are also disabled during the “exercise mode.”

In the effort to minimize the impact upon the timing, and hence the clock rate, the random number generator and all other mode selection logic are placed off the critical path. We merely add a MUX (multiplexer) along the critical paths and an extra input into the DFF enable logic. A post-synthesis timing report of the proposed router micro-architecture indicates the delay increase is negligible. This extra delay is so small that it will be absorbed by the chip’s guardband. As the circuit is used for a longer period and gets older, the additional circuit slows down the aging process, and in turn, it slows the rate of delay increase along the critical path. Therefore, adding this circuit is actually beneficial to timing/clocking in the long-term.

We explore two efficient mechanisms to implement the “Random Gen” component. First as a serial-update, parallel-read LFSR (Linear Feedback Shift Register)² and second as a set of pre-defined vectors, randomly generated at design-time. The LFSR is composed of a 128-bit register, and we use the first 20 bits for the “Request” signal and the following 100 bits for the “Route” signal (eight bits are unused). The taps of the LFSR are placed at 0, 2, 27 and 99th registers such that it has the maximal length period. When “Random Gen” is implemented as a pre-defined set of random vectors, we generate a number of 120-bit random vectors and store them in a small ROM within the router. The quantity of random vectors is decided at design time and we tested a range of such numbers at 1, 2, 4, 8, 16, 32, 64 and 128 entries. This number has implications on the circuit’s energy consumption and lifetime which will be discussed in Section 6.

In order to mitigate the impact on activity factor of the proposed “exercise mode” input, we propose to rotate the input vector once every pre-defined period of time. This is feasible because the duty-cycle is insensitive to the frequency of input vector rotation while the activity factor is linearly related to it. We tested a range of rotation periods, of 8, 32 and 128 clock cycles. We explore the implications of the period length on the circuit’s energy consumption and lifetime in Section 6.

²Note, this is not a traditional implementation of an LFSR; a typical serial-read LFSR was found to consume too much energy to be practical for a random vector of this size. The trade-off being the extra iterations required to achieve pseudo-random data.

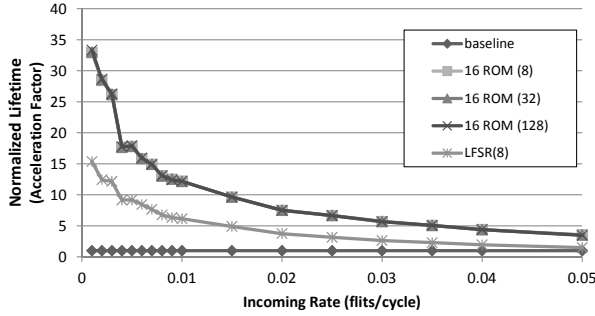


Figure 10: Normalized Lifetime (Acceleration Factor) for router under a given synthetic incoming rate from 0.001 flits/cycle to .05 flits/cycle.

6. EVALUATION

In this section we first outline our experimental setup. This is followed by a detailed exploration of the benefits and costs of our proposed technique.

6.1 Experiment Setup

The baseline router, adapted from RTL code made publicly available by Becker [4], contains three pipeline stages. The detailed parameters of the router are listed in Table 1. Its RTL code was modified to include the “exercise mode” described in Section 5. It is synthesized using Synopsys Design Compiler and mapped to a TSMC 45 nm technology library at 1 GHz. We note that the added exercise circuit is largely off the critical path (see Section 5); thus, router synthesis produced the same clock frequency as the baseline router of 1 GHz. The critical paths were extracted using Synopsys Design Vision. All paths with 10% slack were retained and analyzed. The wire activity (activity factor and duty cycle) along the paths are extracted with Synopsys VCS and analyzed offline. The power consumption is evaluated using PrimeTime.

The router is evaluated under both synthetic and realistic workloads. The realistic workloads are captured as traces from the gem5 simulator [8] emulating a 64-core system executing multithreaded programs from the PARSEC v2.1 suite [7]. Table 1 summarizes the system configuration. We compute incoming rate of each router in 8×8 mesh network’s individually under X-Y DOR routing. The per-router min, max and average incoming rates for each application were calculated. These rates are then applied to the synthesized router to extract the activity of its wires. For both synthetic and realistic workloads, we execute the post-synthesis models of both the baseline and proposed routers, for 100,000 cycles, to measure the wire activity.

6.2 Experiment Results

6.2.1 Aging under synthetic workloads:

We first examine the potential gain in router lifetime of the proposed technique versus baseline for a range of arbitrary incoming rates. We also compare the two kinds of “Random Gen” implementations. As previously discussed, the per-router incoming rate under PARSEC workloads tends to vary between 0.0005 (*x264*) to 0.085 (*canneal*). Figure 10 shows the normalized acceleration factor, calculated according to Equation 13, versus the baseline router at the same

incoming rate. As explained in Section 4, the acceleration factor gives the lifetime of the system under consideration, normalized to the lifetime of the reference system. In Figure 10, “baseline” is the normalized lifetime of the baseline router, which is always 1, while “16 ROM” and “LFSR” indicate the cases when the “Random Gen” is implemented with 16-entry ROM of pre-defined random vectors and with the LFSR, respectively. The number in parenthesis indicates the vector rotation period in cycles.

Figure 10 shows, lifetime improves dramatically for the routers with low incoming rates. Generally low incoming rates cause a greater bias in duty cycle, and hence, more room for the improvement, thus the greatest gains in lifetime occur with the lowest incoming rates. The ROM-based “Random Gen” (“16 ROM (8-128)”) consistently outperforms the LFSR method across all incoming rates. We find that the “LFSR” is outperformed by the “16 ROM” schemes by 2:1 across all incoming rates. The LFSR does not generate as well-balanced, random, 120-bit vectors as the pre-defined ROM. This is primarily because the serial shift, parallel read design of our LFSR results in only one bit of entropy for every 8 cycles in the “LFSR (8)”, while the 16 ROM schemes result in a complete new value every rotation period.

Figure 10 shows no significant difference in lifetime between the three different random vector rotation periods (“16 ROM(8)”, “16 ROM(32)” and “16 ROM(128)”). Another parameter for ROM-base random vector generation is the number of entries in the ROM. In Figure 10, results shown are for ROMs with 16 entries (“16 ROM (8-128)”). We tested ROM sizes between 1 and 128 entries to determine the minimum size required to produce the desired increase in lifetime (figure omitted for brevity). In general, lifetime increases with the number of entries increases, however, at 16 entries (“16 ROM (8-128)”) the acceleration factor benefit saturates. Thus, we use 16 entries with the longest period (“16 ROM (128)”) for the remainder of this paper as this design point implies the lowest overhead in terms of storage per router and activity factor.

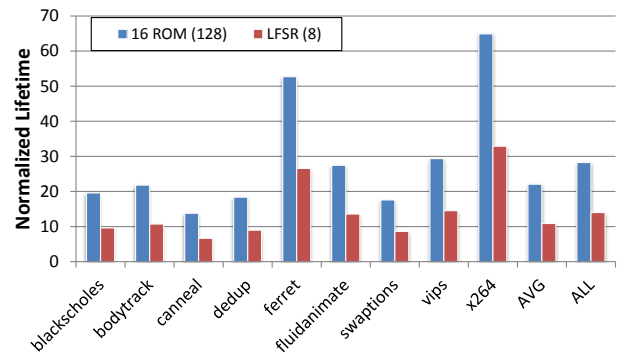


Figure 11: Normalized lifetime of the network using the proposed technique under realistic workload.

6.2.2 Lifetime under PARSEC workloads:

Figure 11 depicts the normalized lifetime of the network using the proposed technique under PARSEC workloads. The lifetime of the network is estimated by computing the acceleration factor of the router with the minimum incoming rate in the network, as it is the most susceptible to aging

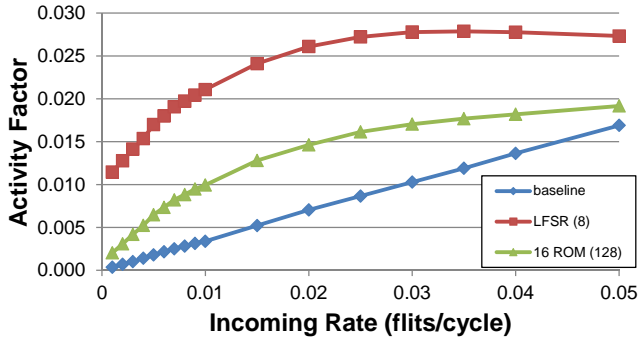


Figure 12: Activity Factor Versus Injection Rate.

effects. The reference system is the baseline router receiving the same incoming rate. For each benchmark, we evaluate the normalized lifetime of the exercise logic using the two different random data generation schemes, which are “16 ROM (128)” and “LFSR (8)”. The “16 ROM (128)” achieves an average of $\sim 22\times$ improvement in lifetime (bars marked “AVG”). As expected, the proposed technique performs better when incoming rate is low. Figure 7 shows “ferret” and “x264” are the applications with the two lowest incoming rates in the PARSEC suite. For those applications, the “16 ROM (128)” achieves a $53\times$ and $65\times$ lifetime extension, respectively³. Even when the average incoming rate is as high as 0.05 flits per cycle (“canneal”), “16 ROM (128)” still achieves the normalized lifetime of $13.8\times$ due to the extreme spread in per-router incoming rates from minimal to maximum seen in that application. The LFSR scheme achieves about half of the lifetime improvement of the ROM scheme.

The bars designated as “ALL” denote a case in which the system executes each of the applications sequentially one at a time. In this case, the improvement becomes $\sim 28\times$. We found that the execution times of “ferret” and “x264” are the longest among the applications, and hence the incoming rate is dominated by those two applications when the suite is executed in such a way.

6.2.3 Activity factor:

One potential downside of a technique that decreases the duty cycle along the critical path is that it could increase the activity factor as well, resulting to potential HCI-induced aging problems. Figure 12 shows the average activity factor along the critical paths at various flit incoming rates for different router models. For the “baseline” router, the activity factor is linearly proportional to the incoming rate as the incoming flits are the only stimuli to the allocator. In the modified routers, the activity factor also increases as the incoming rate grows but it increases slightly more rapidly than the “baseline” case. The growth of activity factor with respect to the incoming rate is more rapid at low incoming rates, as the exercise logic has more opportunity to become active. As the incoming rate increases and the exercise logic misses opportunities to generate a new random vector, the increase in the activity factor slows down. At incoming rates of 0.05 and above, “16 ROM (128)” effectively

³Note that $65\times$ is greater than the normalized lifetime gain shown in Figure 10; this is because the minimum incoming rate of this application is less than the minimum value of 0.001 flits/cycle depicted in Figure 10.

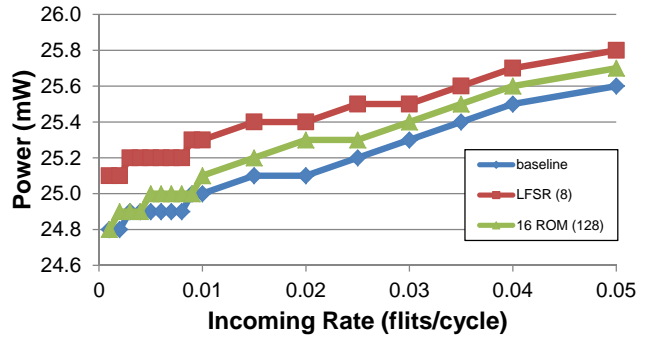


Figure 13: Router Power Consumption Versus Injection Rate (note: Y-axis scaled to provide detail).

matches the activity factor of the baseline router, implying that the proposed technique should not significantly impact HCI for highly loaded routers, where HCI could have a more consequential impact. In Figure 12, “16 ROM (128)” shows less activity factor increase than “LFSR (8)”, this is because the rotation period is less for the LFSR technique; “LFSR (128)” would result in a similar activity factor to “16 ROM (128)”.

6.2.4 Power analysis:

An increase in activity factor in the allocators should be expected to lead to additional dynamic power consumption for the router. Further, the additional “exercise logic” should also require additional static and dynamic power. Thus we performed a power analysis of the baseline and proposed router designs using Synopsys PrimeTime. Figure 13 shows the power consumption with respect to varying incoming rate for the different router models. As expected, router power consumption increases as the incoming rate increases, however, we find that the “16 ROM (128)” technique increases the total router power by less than 1% across all incoming rates. In part, this is because the allocator logic in the baseline router only consumes 1.3% of the total router power, and the additional “exercise logic” only increase the area by less than 1% of the area of original router, limiting the potential for power consumption increase.

7. RELATED WORK

Aging models for transistors have been extensively studied since technology crossed the submicron border, which inherently made the CMOS manufacturing process vulnerable to run-time faults. NBTI and HCI are dominant wearout effects and have thus been more intensively studied [27]. Conventionally, aging effects are studied under DC stress conditions where it is easier to measure transistor parameters [18, 22, 25]. However, AC stress conditions are more realistic for high-frequency long-term CMOS operation; hence works such as those of [19, 36, 27] discuss such long-term models, while other works introduce the relationship between DC and AC stress conditions [37].

The newly emerging devices, such as multi-gate field effect transistors MuFETs and FinFETs [3, 30] at 22 nm process technology and below, have gradually become an object for aging effects exploration. Wang *et al.* [37] make an attempt towards presenting a unified aging model for the effects of both HCI and NBTI. They derive models for double- and

triple-gate FinFETs, for each of these aging effects, that capture specific FinFET geometrical aspects.

Unfortunately, the vast majority of the reported models lack important details, such as values for various constants, measurement conditions, detailed explanation of parameters, etc. Thus, it becomes fairly challenging to employ the existing frameworks, in the context of microarchitecture, to perform meaningful aging effect calculations

Various techniques have been proposed so as to mitigate the aging effect in processor core architectures. Among those proposed mechanisms, Gunadi *et al.* [14] suggested the Colt duty cycle equalizer which balances the duty cycle by alternating true and one's complement data representations. Abella *et al.* [1] introduced the Penelope NBTI-aware processor architecture where they discuss a number of techniques to combat NBTI, including a mechanism which writes special values in memory cells in order to keep the duty cycle at an ideal 50%. Next, Kumar *et al.* [21] proposed periodic cache flipping so as to provide periods of relaxation for the influenced pMOSFET allowing dynamic recovery of the threshold voltage level. The aforementioned three works bear similarities to the "exercise logic" proposed in this paper, in the context that these techniques periodically insert inverted or random values to balance the duty cycle. None of the three, however, deals with NoC router microarchitectures, which are critical to the system's survivability.

Although our approach is similar to the approaches in the aforementioned works, here we actually handle a different problem. In the previous works, the researchers focused on the duty cycle bias on the data paths, while we mainly balance the uneven duty cycles along the control paths. In fact, the sources of the uneven duty cycle are different. In the previous works, it is dominant in biased data contents, while in our work it is more evident during the proved extremely low activity seen in NoC routers (see Section 3). Although the proposed "exercise logic" is also able to resolve the aging problems due to the skewed data content along the data path of the NoC routers, we utilize it only for the timing critical paths where the NBTI impact occurs in its most critical form.

A number of works make attempt to develop a reliability model at architectural level. Srinivasan *et al.*, proposed such a model of a processor core, which considers a set of failure mechanisms [35]. Their assumptions, such as even distribution of failures across failure mechanisms and uniform failure rate across the design, restrict the accuracy of the model when extended to the entire chip. Shin *et al.* further develop this concept [33]. They introduce *effective defect density* and *effective stress condition* coefficients that weigh failure impact across the chip area and run-time respectively. Shin *et al.* illustrated their approach under several failure mechanisms and presented indicative weight coefficients for a set of abstract architectural structures. By contrast, in this work, we examine the actual critical path of a particularly failure-sensitive, critical CMP component, the NoC router. We show that, under realistic workloads, this component is highly susceptible to NBTI-based wear and develop a technique to mitigate this wear.

Aging has been also examined in the NoC domain. Bhardwaj *et al.* [6, 5] propose routing algorithms to mitigate multiple aging mechanisms. They also point out that NBTI plays a major role in NoC router aging, and their routing techniques balance the traffic load across the network to

level-out the aging rates among the routers. The approach is reasonable, in that they force the network traffic to detour through routers of low utilization which, on the contrary, accelerates NBTI-caused aging. However, they use these routing techniques for the opposite effect; the routing algorithms are actually designed to reduce the workload onto the routers which exhibit high utilization, which as we show here are not actually the routers likely to exhibit the most stress-related aging. Fu *et al.* [13] propose a similar technique to ours, in that it inserts special values to idle arbiters to mitigate NBTI. However, they propose this technique to make arbiters less frequently utilized so as to give these routers a chance to recover from the effects of NBTI, which is actually not necessary applicable to frequently utilized circuits. In these previous NoC-oriented studies, it is assumed that the NBTI stress time is proportional to the router utilization, however, on the contrary, we prove that this is not the actual case. Through detailed, gate-level analysis, not found in earlier works, we demonstrated that the duty cycle becomes more skewed when the NoC router is actually under-utilized and not when it is highly- or over-utilized.

8. CONCLUSIONS

The NoC interconnect is critical to the lifetime survival of the CMP system. In this paper, we developed critical path models for HCI- and NBTI-induced wear due to stresses caused by realistic workloads, and applied them onto the interconnect microarchitecture. A key finding from this modeling being that, counter to prevailing wisdom, wearout in the CMP on-chip interconnect is correlated with lack of load observed in the NoC routers, rather than high load. We then developed a novel wearout-decelerating scheme in which routers under low load have their wearout-sensitive components exercised, without significantly impacting cycle time, pipeline depth, area or power consumption of the overall router. We subsequently show that the proposed design yields a $13.8\times$ - $65\times$ increase in CMP lifetime.

Acknowledgments

This work falls under the Cyprus Research Promotion Foundation's Framework Programme for Research, Technological Development and Innovation 2009-10 (ΔEEMH 2009-10), co-funded by the Republic of Cyprus and the European Regional Development Fund, and specifically under Grant $\Delta\text{IE}\Theta\text{NH}\Sigma/\Sigma\text{TOXO}\Sigma/0311/06$.

9. REFERENCES

- [1] J. Abella, X. Vera, and A. González. Penelope: The nbtI-aware processor. In *the 40th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pages 85–96, 2007.
- [2] M. Agarwal, B. Paul, M. Zhang, and S. Mitra. Circuit failure prediction and its application to transistor aging. In *the 25th IEEE VLSI Test Symposium*, pages 277–286, 2007.
- [3] C. Auth. 22-nm fully-depleted tri-gate cmos transistors. In *the 2012 IEEE Custom Integrated Circuits Conference (CICC)*, pages 1–6, 2012.
- [4] D. U. Becker. *Efficient Microarchitecture for Network-on-Chip Routers*. PhD thesis, Stanford University, 2012.
- [5] K. Bhardwaj, K. Chakraborty, and S. Roy. An milp-based aging-aware routing algorithm for nocs. In *the Design, Automation Test in Europe Conference (DATE)*, pages 326–331, 2012.

- [6] K. Bhardwaj, K. Chakraborty, and S. Roy. Towards graceful aging degradation in nocs through an adaptive routing algorithm. In *the 49th ACM/EDAC/IEEE Design Automation Conference (DAC)*, pages 382–391, 2012.
- [7] C. Bienia, S. Kumar, J. P. Singh, and K. Li. The parsec benchmark suite: characterization and architectural implications. In *the 17th International Conference on Parallel Architectures and Compilation Techniques (PACT)*, pages 72–81, 2008.
- [8] N. Binkert, B. Beckmann, G. Black, S. K. Reinhardt, A. Saidi, A. Basu, J. Hestness, D. R. Hower, T. Krishna, S. Sardashti, et al. The gem5 simulator. *ACM SIGARCH Computer Architecture News*, 39(2), 2011.
- [9] J. Blome, S. Feng, S. Gupta, and S. Mahlke. Self-calibrating online wearout detection. In *the 40th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pages 109–122, 2007.
- [10] W. J. Dally and B. Towles. Route packets, not wires: On-chip interconnection networks. In *the 38th ACM/EDAC/IEEE Design Automation Conference (DAC)*, pages 684–689, 2001.
- [11] A. DeOrio, K. Aisopos, V. Bertacco, and L.-S. Peh. Drain: Distributed recovery architecture for inaccessible nodes in multi-core chips. In *the 48th ACM/EDAC/IEEE Design Automation Conference (DAC)*, 2011.
- [12] D. Fick, A. DeOrio, G. Chen, V. Bertacco, D. Sylvester, and D. Blaauw. A highly resilient routing algorithm for fault-tolerant nocs. In *the Design, Automation Test in Europe Conference (DATE)*, 2009.
- [13] X. Fu, T. Li, and J. A. B. Fortes. Architecting reliable multi-core network-on-chip for small scale processing technology. In *DSN*, 2010.
- [14] E. Gunadi, A. A. Sinkar, N. S. Kim, and M. H. Lipasti. Combating aging with the colt duty cycle equalizer. In *the 43rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pages 103–114, 2010.
- [15] J. Howard, S. Dighe, S. R. Vangal, G. Ruhl, N. Borkar, S. Jain, V. Erraguntla, M. Konow, M. Riepen, M. Gries, G. Droege, T. Lund-Larsen, S. Steibl, S. Borkar, V. K. De, and R. V. Der Wijngaart. A 48-core ia-32 processor in 45 nm cmos using on-die message-passing and dvfs for performance and power scaling. *IEEE Journal of Solid-State Circuits*, 46(1), 2011.
- [16] L. Huang and Q. Xu. Agesim: A simulation framework for evaluating the lifetime reliability of processor-based socs. In *the Conference on Design, Automation and Test in Europe (DATE)*, pages 51–56, 2010.
- [17] ITRS International Technology Roadmap for Semiconductors. Process integration, devices, and structures (PIDS), 2009.
- [18] JEDEC Solid State Technology Association. Failure mechanisms and models for semiconductor devices, JEP122G, 2011.
- [19] U. Karpuzcu, B. Greskamp, and J. Torrellas. The bubblewrap many-core: Popping cores for sequential acceleration. In *the 42nd Annual IEEE/ACM International Symposium on Microarchitecture, (MICRO)*, pages 447–458, 2009.
- [20] K. J. Kuhn. Reducing variation in advanced logic technologies: Approaches to process and design for manufacturability of nanoscale cmos. In *the IEEE International Electron Devices Meeting, (IEDM)*, pages 471–474, 2007.
- [21] S. V. Kumar, C. H. Kim, and S. S. Sapatnekar. Impact of nbti on sram read stability and design for reliability. In *the 7th International Symposium on Quality Electronic Design, (ISQED)*, pages 6–pp, 2006.
- [22] X. Li, J. Qin, and J. Bernstein. Compact modeling of mosfet wearout mechanisms for circuit-reliability simulation. *IEEE Transactions on Device and Materials Reliability*, 8(1):98–121, 2008.
- [23] Y. Li, S. Makar, and S. Mitra. CASP: Concurrent Autonomous Chip Self-Test using Stored Test Patterns. In *the Conference on Design, Automation and Test in Europe (DATE)*, pages 885–890, 2008.
- [24] Y. Lu, L. Shang, H. Zhou, H. Zhu, F. Yang, and X. Zeng. Statistical reliability analysis under process variation and aging effects. In *the 46th ACM/IEEE Design Automation Conference, (DAC)*, pages 514–519, 2009.
- [25] E. Maricau and G. Gielen. A methodology for measuring transistor ageing effects towards accurate reliability simulation. In *the 15th IEEE International On-Line Testing Symposium, (IOLTS)*, pages 21–26, 2009.
- [26] S. Nassif, K. Bernstein, D. Frank, A. Gattiker, W. Haensch, B. Ji, E. Nowak, D. Pearson, and N. Rohrer. High performance cmos variability in the 65nm regime and beyond. In *IEEE International Electron Devices Meeting (IEDM)*, pages 569–571, 2007.
- [27] F. Oboril and M. Tahoori. Extratime: Modeling and analysis of wearout due to transistor aging at microarchitecture-level. In *the 42nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 1–12, 2012.
- [28] L.-S. Peh and W. J. Dally. A delay model and speculative architecture for pipelined routers. In *the 7th International Symposium on High-Performance Computer Architecture (HPCA)*, pages 255–266, 2001.
- [29] M. D. Powell, A. Biswas, S. Gupta, and S. S. Mukherjee. Architectural core salvaging in a multi-core processor for hard-error tolerance. In *the 36th Annual International Symposium on Computer Architecture (ISCA)*. ACM, 2009.
- [30] M. Saitoh, K. Ota, C. Tanaka, Y. Nakabayashi, K. Uchida, and T. Numata. Performance, variability and reliability of silicon tri-gate nanowire mosfets. In *the IEEE International Reliability Physics Symposium (IRPS)*, pages 6A–3, 2012.
- [31] T. Sakurai and A. Newton. Alpha-power law mosfet model and its applications to cmos inverter delay and other formulas. *IEEE Journal of Solid-State Circuits*, 25(2):584–594, 1990.
- [32] T. Schonwald, J. Zimmermann, O. Bringmann, and W. Rosenstiel. Fully adaptive fault-tolerant routing algorithm for network-on-chip architectures. In *the 10th Euromicro Conference on Digital System Design Architectures, Methods and Tools, (DSD)*, pages 527–534, 2007.
- [33] J. Shin, V. Zyuban, Z. Hu, J. Rivers, and P. Bose. A framework for architecture-level lifetime reliability modeling. In *the 37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, (DSN)*, pages 534–543, 2007.
- [34] J. C. Smolens, B. T. Gold, J. C. Hoe, B. Falsafi, and K. Mai. Detecting emerging wearout faults. In *the IEEE Workshop on Silicon Errors in Logic-System Effects (SELSE)*, 2007.
- [35] J. Srinivasan, S. Adve, P. Bose, and J. Rivers. The case for lifetime reliability-aware microprocessors. In *the 31st Annual International Symposium on Computer Architecture, (ISCA)*, pages 276–287, 2004.
- [36] B. Tudor, J. Wang, Z. Chen, R. Tan, W. Liu, and F. Lee. An accurate and scalable mosfet aging model for circuit simulation. In *the 12th International Symposium on Quality Electronic Design (ISQED)*, pages 1–4, 2011.
- [37] Y. Wang, S. Cotofana, and L. Fang. A unified aging model of NBTI and HCI degradation towards lifetime reliability management for nanoscale MOSFET circuits. In *the IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH)*, pages 175–180, 2011.
- [38] Z. Zhang, A. Greiner, and S. Taktak. A reconfigurable routing algorithm for a fault-tolerant 2D-Mesh Network-on-Chip. In *the 45th ACM/IEEE Design Automation Conference, (DAC)*, pages 441–446, 2008.