Chapter 14
Design of Reliable NoC Architectures

Noel Daniel Gundi, Prabal Basu, Sanghamitra Roy, and Koushik Chakraborty

14.1 Introduction

Rapid technology scaling has fueled a seismic growth in the number of on-chip resources. To procure an efficient performance throughput, effective communication between the hundreds of cores proves to be a very vital feature. Communication delay in System-on-Chips is a massive determinant in the overall system performance. In order to facilitate the ongoing communication needs between hundreds of cores, Network-on-Chip has been embraced as the *de facto* standard for the on-chip communication, owing to their performance, scalability, and flexibility advantages.

Providing a reliable NoC design has been a challenging task, as the performance of an NoC is primarily based on the network topology and routing algorithm. As the NoC plays an important role in the performance and energy efficiency of the system, addressing the factors affecting the NoC reliability appears to be of prime importance. This chapter focuses on the enhanced design techniques for an NoC architecture with prime stress on addressing factors affecting the reliability of an NoC. Section 14.2 discusses the challenges posing a threat on the NoC reliability. Section 14.3 elaborates the various schemes to tackle the NoC reliability issues. Section 14.4 summarizes the promising design solutions discussed in this chapter.

N. D. Gundi (✉) · P. Basu · S. Roy · K. Chakraborty
Utah State University, Logan, UT, USA
e-mail: noeldaniel@aggiemail.usu.edu; prabalb@aggiemail.usu.edu; sanghamitra.roy@usu.edu; koushik.chakraborty@usu.edu

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2021

P. Mishra, S. Charles (eds.), *Network-on-Chip Security and Privacy*,
https://doi.org/10.1007/978-3-030-69131-8_14

14.2 Factors Affecting NoC Reliability

As an NoC is deployed across the parallel computing environment, multiple issues emerge, which questions the credibility of an NoC design. Reliability of NOC is affected by various factors ranging from the problems arising due to device aging to unbalanced utilization of NoC components. Sections 14.2.1–14.2.6 address the various issues which degrade the performance of the NoC thereby, affecting the entire system performance.

14.2.1 Negative Bias Temperature Instability and Electromigration

Negative Bias Temperature Instability (NBTI) occurs due to the negative bias voltages at higher temperatures creating traps between layers of MOSFETs [1]. NBTI causes a degradation in drain current and absolute increase in the threshold voltage. On the other hand, Electromigration is the process of the transportation of metallic atoms by the electron current flow.

Table 14.1 shows the different schemes considering the varying impact of NBTI and Electromigration on the NoC routers and links. Figure 14.1 depicts the increase of latency with time due to the individual and combined effect of NBTI and Electromigration.

14.2.2 Asymmetric Traffic Utilization

Asymmetric utilization of NoC components significantly exacerbates the aging degradation. Higher utilization in particular NoC components manifests in a power-performance degradation due to rapid aging of these NoC components. Mishra et al. [2] observed that there is up to 2× utilization in the centralized routers in comparison to the peripheral routers. Increase in utilization symmetry in the centralized routers is demonstrated in Fig. 14.2.

Table 14.1 Different degradation schemes

Scheme	Degradation in routers	Degradation in links
A	NBTI	NONE
B	NBTI	NBTI
C	NBTI	Electromigration
D	NBTI	NBTI and Electromigration

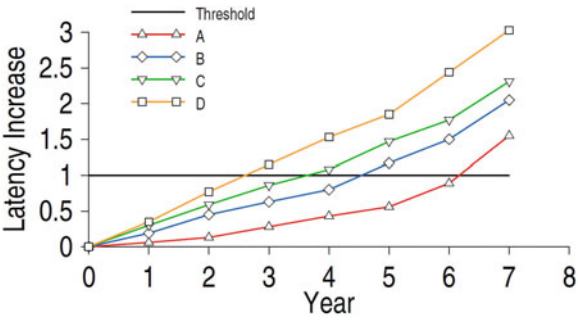


Fig. 14.1 Time taken for the network to become faulty under various aging models (high injection rate)

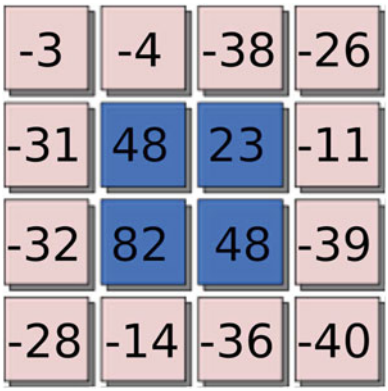


Fig. 14.2 Percentage traffic increase of each router using Buffered-Router Aware Routing (average across PARSEC benchmarks). This utilization difference leads to more than 2× divergence in NBTI induced performance degradation

14.2.3 Hot Carrier Injection

The phenomenon of Hot Carrier Injection (HCI) occurs when a carrier leaves the channel overcoming the potential barrier between the silicon and the gate oxide [3]. Carriers leaving the channel are deposited in the gate oxide region of the transistor. Over a period of time, the conductive properties of the transistor are altered due to the deposited carriers leading to an overall degradation in the threshold voltage, drain saturation current, and transconductance [4–6]. HCI degradation is majorly dependent on the switching activity of the transistors.

Figure 14.3 depicts the switching activity for the gates across an NoC architecture. From Fig. 14.3 it is evident that only 25% of the gates are responsible for 75% of the switching activity. The resulting asymmetry leads to an unbalanced HCI degradation across the NoC architecture leading to an early failure of an NoC.

14.2.4 Quality-of-Service (QoS) Policies

Enforcement of Quality-of-Service (QoS) Policies becomes quintessential to ensure fairness among different users/programs when limited number of resources are

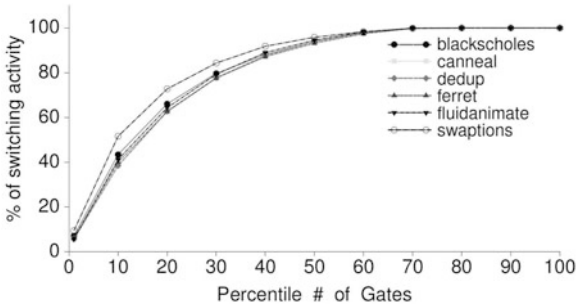


Fig. 14.3 Cumulative distribution function of the switching activity vs gate count

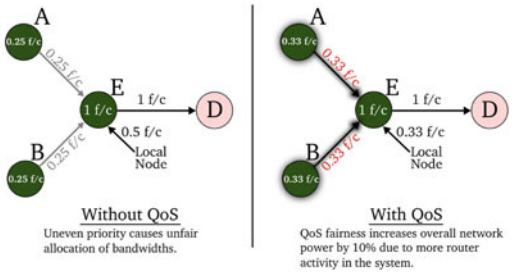
shared by large exascale computing system [7]. However, as NoC is scaled, administering QoS dramatically lowers its Mean Time To Failure (MTTF) due to the increased power consumption and raised thermal profile. The elevated power/thermal characteristics arises due to the balanced resource management provided by the QoS support [8], rather than an increase in performance. Hence, QoS support leads to a wearout acceleration and shortened lifetime even though it offers an identical bandwidth.

Figure 14.4a demonstrates the three nodes A, B, and E attempting to send flits to D. Nodes A and B receive unfair treatment without QoS as they receive only 1/4th of the bandwidth due to contention. Fair distribution of the link bandwidth for all three nodes between E–D link is provided by the QoS support. However, the risen network activity results in an increase in the power consumption which results in wearout acceleration for NoC devices. Figure 14.4b and c demonstrate the effects of QoS support on the power and MTTF of an NoC.

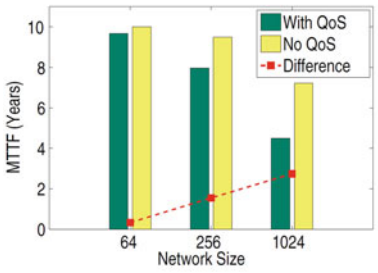
14.2.5 Voltage Emergencies

Voltage Emergencies in an NoC (VEN) arise due to the collaboration of various technology trends. A substantial increase in energy savings can be observed in computation than in communication due to technology scaling. NoCs consume a remarkable proportion (i.e., 36%) of chip power [9]. NoC draws a large current in its circuit components due to its rising power footprint. VENs emerge in the system resulting in timing errors, due to the variations in the current drawn by the NoC. Timing errors[1] generated by VEN can be mitigated by voltage guardbands. However, using guardbands alone can significantly deteriorate the energy efficiency. Timing errors in an NoC router pipeline presents a distinct challenge in comparison to the processor pipeline [10], as pipeline flush and recovery mechanisms cannot be used in the NoC pipeline.

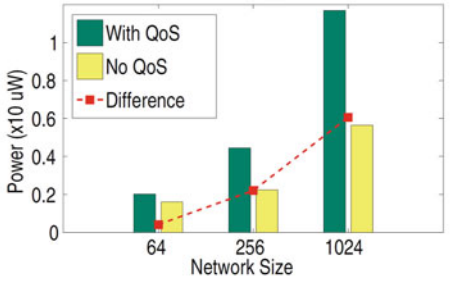
[1] A timing error is observed when the pipe stage delay in exceeds the clock period.



(a) QoS effect on the Network Traffic.



(b) MTTF Impact of QoS Support.



(c) Effect of Providing QoS on the Average Router Power Consumption.

Fig. 14.4 Figure (**a**) and (**b**) shows the conflicting goals of QoS support and sustainability: although the bandwidth offered by the NoC remains unchanged, different resource usage under QoS causes an accelerated wearout and a shortened lifetime. Figure (**c**) shows the effect of providing QoS on the average router power consumption

Fig. 14.5 Frequency of timing errors in the routers of a 8×8 NoC for real world applications

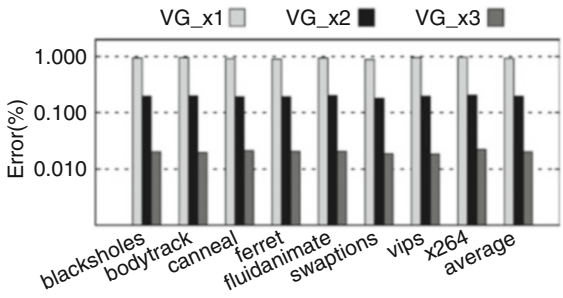


Figure 14.5 depicts the frequency of timing errors in the routers of a 8×8 NoC for the voltage guardbands (VG_x1, VG_x2, VG_x3) set at (22%, 26%, and 30%) above the nominal supply voltage. Timing errors induced from VEN lead to data corruption, flit redirection, and other functional errors. Hence, it is crucial to design energy efficient techniques to handle VEN induced timing errors.

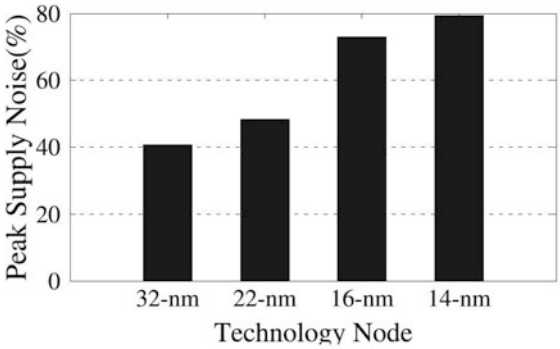


Fig. 14.6 Result is normalized to the corresponding 32-nm technology values. Figure highlights the variation of interconnect circuit parameters per unit length

14.2.6 Power Supply Noise

Modern multiprocessor system-on-chips (MPSoCs) encounter a rising concern due to the integrity of supply voltage. Switching of logic devices due to the uneven distribution of current results in the emergence of noise in Power Delivery Network (PDN), leading to a drop in the supply voltage. The performance and energy efficiency of the system components is severely affected by the Power Supply Noise (PSN). Additionally, scaling of technology node further exacerbates the problem due to the decreasing size and higher device density.

Sources of voltage noise in a PDN are: resistive drop (IR) and inductive drop ($L(\Delta i/\Delta t)$). Voltage drop across the resistances of the power delivery wires causes IR drop, which is proportional to the current (I) in the circuit. Inductive drop, on the other hand, is caused by the wire inductance (L) of the power grid and is proportional to the rate of change of current through the inductance. Figure 14.6 depicts the trend of RLC parameters at smaller technology nodes. Figure 14.6 shows that, the peak noise increases from 40% of the supply voltage at the 32-nm technology node to about 80% of the supply voltage at the 14-nm technology node, if the power distribution strategy remains unchanged.

14.3 Reliable NoC Design Methodologies

Overcoming the reliability problems requires a profound understanding of the intrinsic architecture details, which in turn can be utilized to procure a feasible solution. Additionally, understanding whether the problem can be mitigated or whether the effects of the problem can be delayed proves vital in the direction of developing a reliable design. For example, NBTI (Sect. 14.2.1) is critical, but a recoverable device aging mechanism. However, HCI (Sect. 14.2.3) is an unrecoverable aging phenomenon [11]. To restore the impacts of the factors affecting an NoC design discussed in Sect. 14.2, variety of strategies based on the investigations from

innovative research [12–17] will be explored in this section, in addition to various concurrent research works (Sect. 14.3.7) in this field of work.

14.3.1 Overcoming NBTI and Electromigration

To tackle the problem of NBTI and Electromigration, balancing of the network traffic is essential. Balancing of the asymmetric network utilization can be achieved using a reliability metric and utilizing this metric in an aging-aware adaptive routing algorithm.

The reliability metric is determined based on the intensity of traffic a stressed router/link can handle. Hence the reliability metric *TTPE* is defined as the fraction of the nominal traffic that a stressed router/link should accept during a particular epoch [12]. Significance of the *TTPE* for an aging-stressed NoC design is based on the following facts:

1. *TTPE* determines an upper limit on the amount of traffic that a router or link should accept so as to keep the variation in network latency below a pre-defined threshold for a particular aging period.
2. *TTPE* is derived from continuous monitoring of the traffic, and is used to adapt the routing policies for every epoch to mitigate the long-term degradation in the NoC.

TTPE varies over the runtime with different values during different epochs for each stressed router and link.

The calculation of *TTPE* involves the following stages:

- **Threshold calculation:** The congestion-aware routing algorithm that routes the flits based on both local and global congestion information is profiled. The total time taken to route these flits is then divided into several epochs. The significance of adding epochs lies in the fact that an application's communication characteristics may change during the runtime and therefore the traffic must be monitored continuously. This process keeps track of the link and the router utilization during runtime and takes additional measures if the utilization reaches *TTPE* for the epoch under consideration. For each epoch, the n most stressed links and routers are considered based on their utilization. Based on the NBTI and electromigration of these stressed links and routers, the *TTPE* is calculated.
- **Using *TTPE* Estimation in Routing:** The computed *TTPE* for different epochs is stored in the form of lookup tables (SL_{set}) in each router. The router at runtime can then select the appropriate *TTPE* depending on the epoch. During this stage, the routing tables for each router are computed. In order to minimize network latency and communication energy, only the deadlock-free shortest paths for each flow are selected.

The routing algorithm involves the following two stages (Algorithm 1):

Algorithm 1: Aging_Adaptive

For each flow,
1. Select the best shortest path from the routing table which:
   a) suffers from least delay variation due to aging (sc_{age} is minimum).
   b) is least congested based on global and local congestion information (sc_{cong} is minimum).
2. For each stressed link in SL_{set} of each epoch:
   a) Check if the link meets its $TTPE$:
      - If the link has already reached its $TTPE$, keep the link idle for the rest of the epoch (insert recovery cycles).
      - If link utilization is safely below its $TTPE$ then there is no need for inserting recovery cycles.

1. **Congestion and aging-aware routing:** For each flow at runtime, the routing algorithm selects the best shortest path from the routing table that (i) suffers from least aging degradation i.e. the path that suffers from least delay variation due to aging (1-a); and (ii) is least congested (1-b). Higher priority is given to a path that least degraded as compared to a path with the least congestion.
2. **Honoring *TTPE* by employing recovery cycles:** During the execution of the routing algorithm, each stressed link in SLset is checked to see if it meets its respective *TTPE* for every epoch (2-a). There can be two possible cases: (i) In the epoch, if the link has already reached its *TTPE*, then the link must be kept idle for the rest of the epoch so that its utilization does not exceed its *TTPE*; and (ii) If the link operates safely inside its *TTPE* for that epoch, then there is no need for inserting idle cycles. The physical significance of inserting these idle cycles is that they provide additional time to the links and routers to recover from the aging stress. Therefore, these additional idle cycles are called as recovery cycles. This procedure also avoids unnecessary insertion of recovery cycles in the epoch and thus keeps the network latency in check.

14.3.2 Balancing Traffic Utilization

Balancing of the traffic utilization can be achieved by exploiting the criticality of the various flits in the NoCs [13]. The health of the routers in the network is tracked using a Wearout Monitoring System (WMS). The WMS and the criticality information are used to implement an aging-aware routing schemes.

14.3.2.1 Criticality of Different Flits in NoCs

The latencies of various packets transmitted through an NoC can have varied effects on performance. Previous works have exploited this criticality to improve system performance [18, 19].

Criticality Classification

In general, precise estimation of the packet criticality at the NoC router is hard as it merely has information about source–destination and the packet type. A thorough criticality estimation may require information about the relative performance of running program threads [18, 20], detailed cache coherence transitions, and so forth. To mitigate this complexity, a low-complexity approach is employed, which requires no change in existing interfaces. This involves identifying criticality based on packet type and source–destination. Table 14.2 shows the summary of classification. Using this policy, data packet transmitted from L1 to L2 (destination) is tagged as non-critical in a shared two level cache hierarchy. A vast majority of these packets are writebacks because of cache eviction, and thus the system performance is insensitive to their network latency. Some of these packets are also a result of data sharing among on-chip cores, but these are expected to be a much smaller component due to the predominance of private data even in multi-threaded programs [21].

Figure 14.7 shows the percentage of non-critical packets of PARSEC benchmarks averaged across all the buffered routers. An average of 49% of packets traversing through the buffered routers are non-critical and can actually take a different routing path with minimal performance degradation. Moreover, all benchmarks show substantial opportunity, ranging from 44% to 51% in these benchmarks. By redirecting non-critical traffic to the bufferless routers, the utilization of the buffered routers is minimized, thereby mitigating the aging effects in the buffered routers.

Table 14.2 Packet criticality classification

Data messages		
Source	**Destination**	**Classification**
L1 Cache	L2 Cache	Non-critical
L2 Cache	L1 Cache	Critical
Memory	L2 Cache	Critical
L2 Cache	Memory	Non-critical
Control Messages		
Source	**Destination**	**Classification**
L1 Cache	L2 Cache	Critical
L2 Cache	L1 Cache	Critical
Memory	L2 Cache	Critical
L2 Cache	Memory	Critical

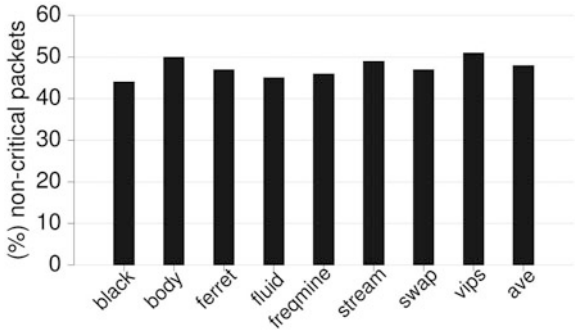


Fig. 14.7 Percentage of non-critical data packets routed through the buffered routers

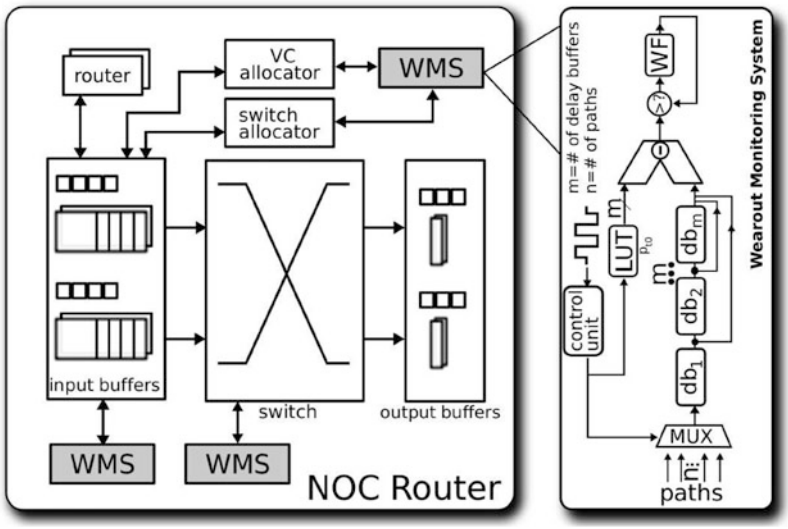


Fig. 14.8 WMS circuit. Each path delay is sampled through a buffer sequence and compared with the reference delay to calculate the WF

14.3.2.2 Wearout Monitoring System (WMS) for NoC Routers

To be able to guide the aging-aware routing algorithm, the WMS profiles the extent of degradation in each router. The WMS circuit shown in Fig. 14.8 augments all pipeline stages of a router. As the performance degradation of a router is dictated by the worst case delay degradation in any pipeline stage, the monitoring system measures the maximum delay degradation across all paths in different pipeline stages. Within a stage, the WMS uses a multiplexer to estimate the delay of all n paths in a combinational logic. The control unit in Fig. 14.8 alters the multiplexer select signal in each cycle to choose which path to measure. Then, a series of m cascaded delay buffers (db_1, db_2, …, db_m) sample the signal at equal time intervals. The state transition captured at the output of each delay buffer provides

an estimate of the delay of the path. Finally, the comparator selects the maximum delay degradation among the n paths over a span of n cycles. The WMS measures the Wearout Factor (WF) as follows:

$$WF_{router} = \max(wf_1, wf_2, \ldots, wf_N) \tag{14.1}$$

$$wf_i = \max(wf_{p1}, wf_{p2}, \ldots, wf_{pn}) \tag{14.2}$$

where, $wf_1, wf_2, \ldots, wf_N$ are the wearout factors for N stages of the router micro-architecture, and $wf_{p1}, wf_{p2}, \ldots, wf_{pn}$ are the wearout factors of the n paths in a single stage i.

14.3.2.3 Criticality-Driven Path Selection

The criticality-driven routing incorporates two major design considerations:

1. Criticality of the incoming packet.
2. WF that dictates the current aging.

The maximum threshold for deflecting non-critical packets is defined as DFL_{Max}. Subsequently, based on the aging degradation in a router, the defection rate is pro-rated in that router.

Integrating Criticality in Routing

To drive the deflection logic in the routing path selection, the source router adds a single bit to store the criticality in the header flit of every packet. All intermediate routers peek into this criticality bit to select different routing paths based on criticality.

Integrating Wearout Monitoring

Different routers can undergo different aging degradation based on their utilization history. In a given router, the WF provides its current aging degradation. Table 14.3 shows the pro-rating scheme used in this work. For example, a router with a WF

Table 14.3 WF based deflection estimation

Wearout factor range	Scheme
0.00–0.50	$\frac{1}{8} \times DFL_{max}$
0.50–0.75	$\frac{1}{4} \times DFL_{max}$
0.75–1.00	$\frac{1}{2} \times DFL_{max}$
1.00−+∞	$1 \times DFL_{max}$

of 0.8 will deflect 25% of all non-critical packets, assuming DFL_{Max} is 0.5. At every sampling interval of the WMS, the WF will be sent to adjacent routers to communicate the degradation of a particular router and a corresponding link. Each router stores the WF of four adjacent routers (North, South, East, West) in dedicated WF registers.

Deflecting Non-critical Packets

For every incoming flit in a router, the deflection logic uses the WF and packet criticality information to determine whether the packet will be sent in the direction of the pre-established path or deflected away from the buffered router. For a bufferless router, this task is accomplished by using a multiplexer and a selection logic. For a buffered router, an additional entry is added in the routing table corresponding to the possible deflection paths for each output port. For instance, an output in the North direction can be deflected to East or West if it is coming from the South input. This logic is accomplished using a 4-bit XOR of the number of ports (N,S,E,W) and the ports used for input and the desired output. Since there can be multiple deflection paths, the one that has no pending flits in the output buffer is used. For ties, the first port using a standard priority encoder is utilized.

14.3.3 Tackling HCI

HCI degradation can be handled by distributing the switching activity across the NoC. The following four techniques are explored in the router micro-architecture: Bit Cruising (BC); Distributed Cycle Mode (DCM); Crossbar Lane Switching (CLS); and BCCLS that is a combination of schemes BC and CLS [14].

14.3.3.1 Bit Cruising (BC)

Bit Cruising interchanges the different portions of the data being transmitted in the crossbar. Bit Cruising is largely motivated by two properties of the programs.

1. Most data in the cache line are aggregated at the lower bits. Hence, most data traversing through the NoC does not occupy the complete channel width of the network. In some cases, all data bits are actually zero.
2. Control requests sent as a single flit do not store information in the most significant portions of the channel as routing information can fit in the first few bytes of the whole channel. The control flit only utilizes 25% of the channel width, leaving the remaining 75% constant [14]. These two characteristics radically lower the switching activity in certain bits while emphasizing others.

To prevent the asymmetry in HCI degradation, the data being sent across the network must be such that the switching activity across the channel is distributed. Passing different data values each time a gate is used will balance the switching activity and uniformly degrade all gates. Hence, the highly changing bits are being circulated around the channel. The Bit Cruiser circuit will be situated in the Network Interface (NI) and does not add any overhead in the critical path of the pipeline of an NoC.

14.3.3.2 Distributed Cycle Mode (DCM)

The Distributed Cycle Mode balances out degradation of transistors by latching an input value in the crossbar during idle times such that unswitched transistors in previous cycles will transition and experience equivalent aging. This scheme does not relieve any HCI aging compared to other schemes but can be beneficial as equally aged transistors have smaller leakage power.

14.3.3.3 Crossbar Lane Switching (CLS)

Another asymmetrical degradation also occurs in the crossbar lanes that are immune to techniques applied in the channel level. This type of asymmetric degradation arises when some input–output pairs are used more than others. This occurrence is demonstrated with an example in Fig. 14.9 where there are two paths (p0 and p1) that both use the same East output port. If path $p0$ is used more than $p1$, then the transistors along the path $p0$ will be sensitized more and hence, experience more HCI degradation. CLS is situated at the frontend of the router pipeline and aims to balance the usage of the crossbar lanes. In the canonical router model, an input port directly forwards flits to the output ports by establishing a physical connection between the two via the crossbar switch. As such, flits coming from the same input port will always use the same crossbar lane to connect to different output

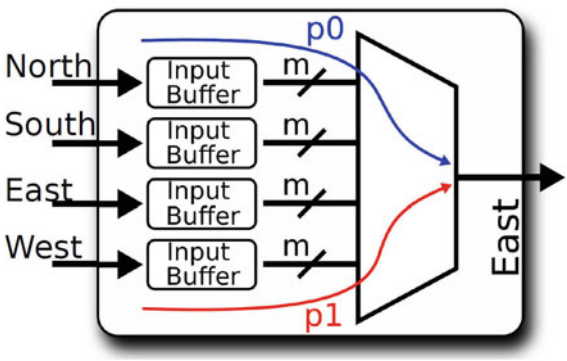


Fig. 14.9 East section of A crossbar switch. CLS works on the inter-lane (by changing the path of the data) level while BC works only on the intra-lane level (by changing the bit ordering within a path)

ports. However, the introduction of Input Buffers (IB) and Virtual Channels (VC) in modern router architectures decouples this one-to-one association because the flits are first stored in the IB before being transmitted to the output ports. With trivial modifications in the VC allocator and the Route Calculation part of the pipeline, it is possible to control the crossbar lane, which an input port will utilize at any given time. This new allocation and routing policy will now cause the crossbar circuit to use a different path and activation circuit, but still send the same data as if it were coming from the original input port. Thus, the correctness of the flit and the route is preserved. Similar to the Bit Cruising technique's cruise setting, CLS will need a knob input to indicate the new mapping between input ports and crossbar lanes.

14.3.3.4 Bit Cruising and Crossbar Lane Switching (BCCLS)

Bit Cruising and Crossbar Lane Switching (BCCLS) is a combination of the BC and CLS schemes. BCCLS combines both the benefit of switching distribution inside a channel (BC scheme) and the distribution of activity across many channels (CLS scheme). The implementation of BCCLS comes naturally because both BC and CLS tackle different portions of the router circuit. BC reshuffles the data sent through the network while CLS effectively changes the port a flit is coming from by modifying the VC allocation and route calculation.

14.3.4 Managing QoS support

Wearout degradation due to a QoS support in an NoC can be managed [15] using a three-step approach as follows:

1. Device level wearout of routers and links is monitored using NoC Health Meter.
2. The wearout information is communicated across the NoC.
3. The wearout information is utilized during NoC routing to dynamically mitigate the effects of aging.

14.3.4.1 NoC Health Meter (NHM)

The NHM profiles the level of degradation in each router and incoming links. The pipe stages of a router is augmented by the NHM circuit as shown in Fig. 14.10, NHM measures the delay degradation in the combinational circuit between two pipeline registers by measuring the slack in each stage. A high resolution all-digital, self-calibrating time-to-digital converter (HR-TDC) consisting of a Vernier Chain (VChain) circuit that has a measurement resolution of 5 ps [22] is used by the NHM to measure the slack. HR-TDC is an in situ delay-slack monitor consisting of a Vernier Chain circuit with an overall measurement window of 150 ps, which

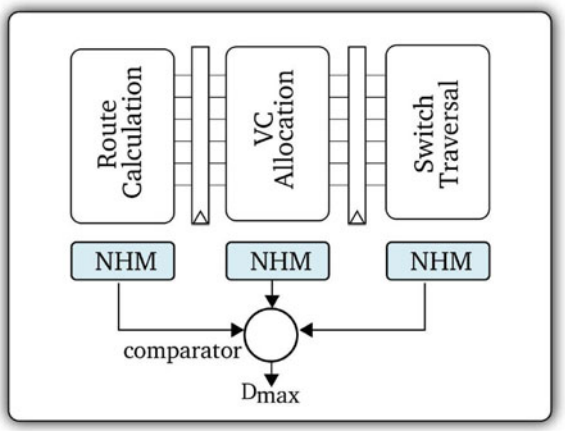


Fig. 14.10 NoC router augmented with NHM

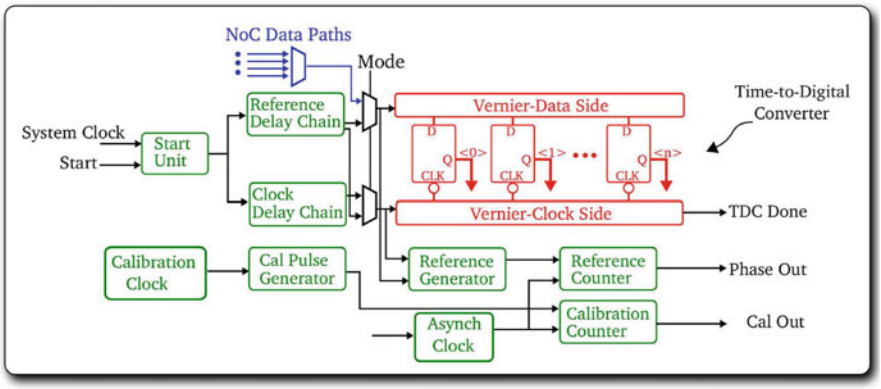


Fig. 14.11 High resolution in situ delay-slack measurement from Fick et al. [22]

is sufficient for timing slack measurements in 2 Ghz+ systems. After measuring the delay degradation of each stage, D_{max}: the maximum degradation among all pipe stages is estimated. Fick et al. has demonstrated that a complete full self-calibration of an entire TDC implemented on a 64-bit Alpha processor can take only 5 min [22].

HR-TDC in NoCs

Usage of HR-TDC circuits to measure the slack or propagation delay of each pipeline stage in an NoC is important because exascale chips with thousands of nodes can experience both global and local Process–Voltage–Temperature (PVT) variability. HR-TDC operates in three modes:

1. **Normal operation**: HR-TDC is measuring the delay fed from the NoC Data Path. Delays of only 30% of the top most critical paths are measured, as measuring all paths is expensive [23]. Data for the Time-to-Digital converter will be aggregated by the NHM to decide the maximum delay among all the pipeline stages.
2. **Reference Delay Chain (RDC) Calibration**: HR-TDC measures the delay of the "Reference Delay Chain" using statistical sampling. Before VChain calibration starts, calibration of the RDC has to be completed.
3. **Vernier Chain Calibration**: HR-TDC calibrates the Vernier Chain in order to maintain a delay of 5ps in each stage of the chain. Eight firmware-controlled capacitor loads are used to make a stage in the VChain tunable, with each load designed to introduce 1 ps shifts in the delay.

Vernier Chain (i.e. red portion of Fig. 14.11) is responsible for measuring the slacks from the NoC data paths in each pipeline stage and converting it to a digital code.

14.3.4.2 Propagating Delay Information and Routing Table Update

The encoded delay information is estimated and propagated through the firmware during the system boot-up, once a month by performing the following three steps :

1. All nodes estimate their D_{max} in parallel throughout the system.
2. D_{max} is broadcasted through the flit link network. To avoid extreme flooding, the network is divided into small equally sized regions. Then, one node from each region broadcasts its D_{max} throughout the system.
3. The routing tables in each node are updated using this D_{max} information.

14.3.4.3 Routing Algorithm

The routing algorithm profiles all two-turn minimal paths of all source–destination pairs. The paths are chosen based on a particular metric such as average router degradation or maximum router degradation. The path for a particular source–destination pair is updated once per month. Figure 14.12 shows an example of our routing algorithm in action. The firmware has already decided which turns to make for a flit with a source–destination of 0 and 11, respectively. The turns are made on nodes 2 and 10. Additionally, a single bit in the head flit is used to indicate which direction the flit should first go, X or Y direction. Whether it is up/down or left/right will be decided by the algorithmic routing based on the relative address of the source and the turning points. Once the flit hits one of the turning nodes, it is going to turn towards the direction of the destination. The algorithm is very scalable because no matter what the size of the exascale NoC is, the routing information stored in a flit (i.e. address of turning points) to be sent from a node to another will only grow by log(n) with n being the number of nodes.

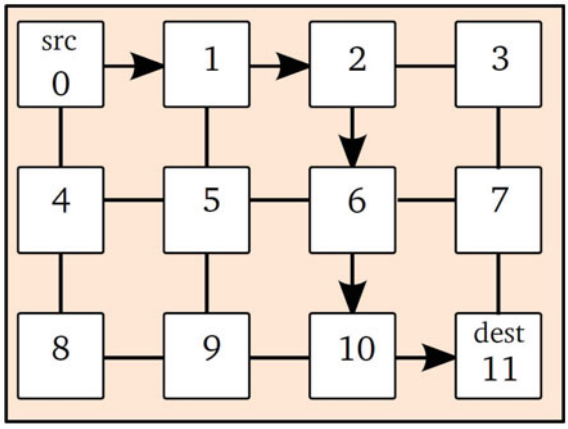


Fig. 14.12 Two-turn path routing

Deadlock Avoidance

Routing packets using various two-turn path configurations can lead to protocol deadlock when cyclic resource dependencies exist. One Virtual Channel (VC) is allocated in each port as an escape channel only to be used when avoiding a deadlock. Normally, when there is no contention, the flits will be routed on the non-escape channels. However, when all non-escape VCs from all routers are occupied for a certain period of time, a cyclic dependency could exist. This is possible because the flits are not restricted to use the same VC ID in each hop in order to maximize the bandwidth of the network. This cyclic dependency is broken by halting further injection in the NoC and allowing in-flight flits to arrive at their destination using deterministic routing via the escape channels.

14.3.4.4 Applying NoC Health Meter in Dynamic Wearout Resilient Routing

NoC health meter can be harnessed by the routing algorithm in two unique ways to dampen QoS-induced traffic stress in NoC routers. Duato's theory is used to restrict virtual channels to specific packet classes to avoid deadlocks [24]. The two algorithms are explained below:

1. **Fresh Routing (FR)**: This algorithm always routes the flits using the least degraded path. This path is constructed by considering several minimal paths and comparing the average wearout information in each path.
2. **Latency Reclamation routing (LR)**: This algorithm seeks to balance congestion and reliability objectives by using dynamic runtime information when deciding a path. LR first compares the number of available credits—a metric quantifying the level of congestion in a node of neighboring routers. If the least degraded path is congested, LR will choose the non-congested path.

The two variants each of these two algorithms are elaborated considering the routing path with p routers, having maximum delays $D_1, D_2, \ldots, D_p$, respectively.

- FR_{Avg}: This scheme uses the average wearout of all routers in a path to select the least-aged path. ($D_{path} = avg(D_1, D_2, \ldots, D_p)$).
- FR_{Max}: This variant of the FR algorithm selects the least-aged path using the maximum router wearout of each path. ($D_{path} = max(D_1, D_2, \ldots, D_p)$). This scheme seeks to limit the wearout of the most degraded router at any time interval.
- LR_{Avg}: This scheme is similar to FR_{Avg}, selecting the least-aged path based on average. However, during congestion, it avoids queuing delay by sending flits in the direction with more credits at times, when the least-aged path is overly congested.
- LR_{Max}: This variant of the LR algorithm also allows credit-based exceptions to the least-aged path. However, like the FR_{Max} scheme, it determines the least aged path using the maximum router delay in each path.

14.3.5 Voltage Emergencies

A reliable design to tackle Voltage Emergencies [16] will comprise of two key parts:

- Error detection and confinement system.
- Recovery mechanisms used to recover corrupted flits.

14.3.5.1 Error Detection and Confinement

VEN induced timing errors are detected at the NoC router pipeline registers using shadow flip-flops [10]. The shadow flip-flops use a delayed clock, allowing double sampling of the combinational logic output. A discrepancy between the sample data in the regular flip-flop and the shadow flip-flop indicates a timing error. Inserting shadow flip-flop is relatively straightforward in an NoC router, as the circuit path in a router pipeline is more uniform in comparison to a typical processor pipeline. Figure 14.13 outlines the circuit-level modifications in an NoC router with 4 pipe stages: input buffer/route calculation, VC allocation, switch traversal, and output buffer. Once an error is detected, restoring error-free communication can only proceed after the error is confined within the router pipeline. On the detection of error, the error has to be confined within the route pipeline, to restore the NoC to error-free communication state. As a traditional NoC pipeline cannot stop a flit from transmission after it has reached the switch traversal stage, two strategies for error confinement based on the error location are explored:

1. ***Error before switch traversal***: Mark the VC as free and increase the credit for the specific port to block the flit before switch traversal. The corrupted flit is

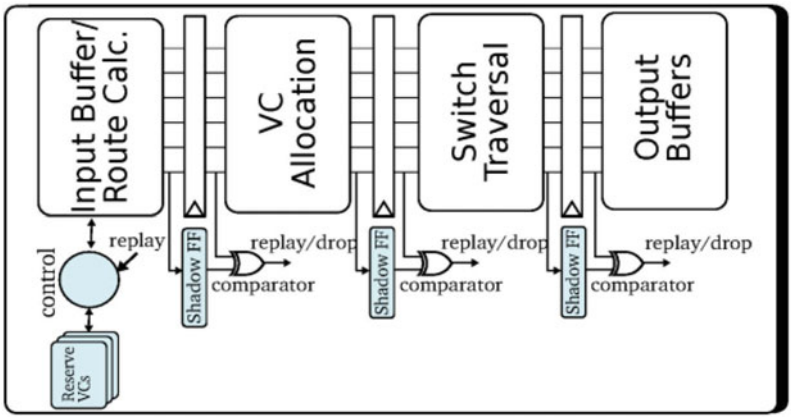


Fig. 14.13 Error detection, confinement, and SRE

overridden, as the new flip is allocated to the free VC entry in the subsequent cycle.

2. ***Error during switch traversal***: Add a poison bit to every output buffer entry. Poison bit is set, when an error is detected on a flit during switch traversal. Therefore, the link traversal is revoked for the particular flit in the next cycle and the buffer and poison bit are cleared to reclaim that entry.

14.3.5.2 Recovery Mechanisms

Two variants of the design based on the tradeoff in performance and complexity overhead are explored.

1. **Router Temporization (RT)** is a low-complexity source-based recovery technique that relies on flit re-transmission.
2. **Selective Router Echo (SRE)** is an in situ dynamic recovery mechanism with a low performance overhead.

Router Temporization (RT)

Router Temporization uses a combination of flit re-transmission and temporary frequency scaling to implement error-free communication in the presence of VEN.

- **Re-Transmission**: The NoC router checks the source for the acknowledgment (ACK) packet to verify the receipt of the data at the destination. The router assumes that the flit has been dropped if the ACK packet is not received after a set amount of time and sends the same flip again until an ACK packet is received.

- **Frequency Scaling**: As the threshold of dropped flits is exceeded, the frequency is lowered (i.e. frequency is halved) to prevent the continuous corruption of flits. VEN typically lasts for a short time span [25]. If the errors persist, the frequency will be consequently lowered until the errors stop. Once the errors stop, the original frequency will be restored using an exponential back-off algorithm.

Selective Router Echo (SRE)

Selective Router Echo is an error recovery system embedded in the NoC router pipeline. In SRE, the router micro-architecture is augmented to mimic a processor pipeline. Figure 14.13 shows the pipeline for the SRE-enabled router. Extra virtual channels are added in the router, called Reserve VCs (RVCs) to keep a record of all in-flights flits which have crossed the input buffer stage. RVCs will replay the erroneous flits in the pipeline in the event of a VEN.

The steps involved in the recovery mechanism are:

- **Stall**: In the case of a VEN induced timing error, the router is stalled and incoming flits to the router are temporarily delayed.
- **Restart**: The router is restarted after stall completion. The delayed flits in the input buffers are permitted to pass through, as the input buffers are cleared to enable the recovery of flits from the RVCs.
- **Restore**: The entries from the RVCs are restored to the input buffers thereby, restoring the router to an earlier state.
- **Resume**: The credit restrictions are lifted and the flits in the input buffer are sent to the targeted output buffers thereby, resuming the normal operation of the router.

14.3.6 Power Supply Noise

PSN can be tackled using flow-control protocols and routing algorithms. The design of a PSN-aware flow-control (PAF) involves a hierarchical approach to dictate the Maximum Current Load (MCL) across the NoC, while ensuring a minimal performance impact [17]. The flow-control information is then utilized in a PAF-aware routing algorithm to tackle PSN.

14.3.6.1 Hierarchical MCL Allocation

High concurrent switching of proximal regions is avoided by carefully adjusting the MCL allocated to each region. To realize the MCL allocation principles at different granularities, a metric Flit Acceptance Potential (FLAP) is defined. For a given input channel of a router, the FLAP is set to 1 when it can receive an incoming

flit (otherwise it is set to 0). For a router, the FLAP indicates the aggregate FLAP of its input channels. Similarly, the FLAP of a particular region represents the aggregate FLAP of the routers in that region. At any given time, the FLAP of a router employing wormhole flow control in a 2-D mesh with four input channels is 4, when all of its input channels can receive at least one flit. The PAF allocates variable MCL to each region by dynamically throttling their FLAPs, irrespective of the space availability in the input channel's buffers. MCL allocation is a hierarchical process that can be applied at multiple spatial granularities. For example, a large region consists of many smaller subregions. The allocated MCL for the large region is distributed among the subregions, ensuring that proximal subregions are not simultaneously allocated with high MCLs. At the lowest granularity, each router's FLAP is managed in a manner that is consistent with the MCL allocation of the entire subregion.

14.3.6.2 Optimizations of PAF

The generic PAF technique needs multiple optimizations to efficiently tackle the design challenges.

Minimizing Performance Impact

Complementary approaches are explored to retain a high performance in the PAF.

- **Judicious FLAP management**: To avoid a large flit delay in a given region, the PAF allows intermittent high and low FLAPs in a router.
- **Topological awareness**: The PAF can be adapted based on the network topology and expected traffic pattern. For example, central routers in a mesh typically experience a high resource demand. This demand can be met by allocating greater FLAPs to the central routers.
- **Congestion awareness**: Two broad classifications of the PAF are explored (Sects. 14.3.6.2 and 14.3.6.2).

Congestion-Agnostic PAF

This variant of PAF statically allocates high and low FLAPs to the regional routers based on a round-robin fairness scheme. The FLAP allocation policy is not influenced by the network buffer occupancy.

Congestion-Aware PAF

This variant of the PAF manages the FLAP allocation based on the relative congestion of the network buffers. The following two congestion awareness at different granularities are considered.

1. **Channel granularity**: The FLAP of the least congested channel of a router is set to 1, so that it can always receive an incoming flit. The other channels' FLAPs are dictated by the aggregate FLAP of the router.
2. **Router granularity**: The least congested router of a region is allocated with a high FLAP. However, the other routers are allocated with low FLAPs to avoid high simultaneous switching. The aggregate FLAPs of the routers are consistent with the allocated MCL of the region.

Avoiding Starvation

Repeated blocking of the flits at the same input channel of a router in successive cycles can cause a starvation. To avoid starvation, the PAF adopts a round-robin fairness scheme to restrict flit reception across all the input channels of a router. Moreover, the PAF uses deterministically routed escape VCs, allowing all the possible turns without a deadlock situation.

Scalability

The PAF is a hierarchical technique that uses local network information at the smallest regional granularity to ascertain the FLAPs of the routers. As the size of the smallest region remains the same even for a larger NoC, the PAF can scale efficiently with the network size.

14.3.6.3 PAF-Aware Adaptive Routing Algorithm

Dynamically throttling the FLAP of a router may cause an intermittent upsurge in the local PSN due to an increased resource contention. This upsurge is circumvented using a PAF cognizant routing algorithm—PAR, which steers the flit toward an unthrottled downstream path. Figure 14.14 depicts the conceptual overview of the PAR. PAR primarily makes the routing decision based on the relative regional congestion information, aggregated solely along the minimal paths. If the chosen output channel has a throttled FLAP, the PAR reroutes the flit to an orthogonal output channel, strictly maintaining the minimal path constraint. This strategy reduces local current spike and the PSN by relieving router contention, but may occasionally increase the network latency by routing some flits toward more congested downstream paths. In a scenario, where both the minimal paths are

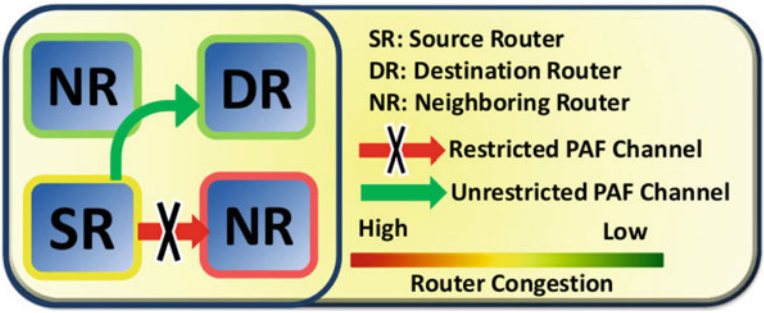


Fig. 14.14 PAR algorithm

blocked due to throttled FLAPs, the flit adheres to the initial channel assignment and waits in the upstream router for another cycle. The PAR incurs no additional circuit overhead as it utilizes the same information required for the PAF.

14.3.7 Concurrent Research Works

In addition to the methodologies discussed through Sects. 14.3.1–14.3.6, cutting-edge research contributions have also been made towards achieving an enhanced NoC design which further stresses on the impact of the reliability threat posed by the issues addressed in Sect. 14.2 [26–35].

14.4 Summary

Increasing performance needs have led to a rapid deterioration of the components in the communication network (NoC). A major cause of this degradation has been the asymmetric utilization of the network components due to device characteristics, resource allocation policies, and uneven traffic flow. In order to restore the reliability of an NoC infrastructure, unique solutions have been explored to mitigate the impending issues. The in situ solutions aid in increasing the lifetime of an NoC and contribute towards the overall system performance.

References

1. D.K. Schroder, Negative bias temperature instability: what do we understand? Microelectron. Reliab. **47**(6), 841–852 (2007)
2. A.K. Mishra, N. Vijaykrishnan, C.R. Das, A case for heterogeneous on-chip interconnects for CMPs, in *ACM SIGARCH Computer Architecture News* (2011), pp. 389–400

3. K. Bhardwaj, K. Chakraborty, S. Roy, An MILP based aging aware routing algorithm for NoCs, in *Proceedings of the IEEE/ACM Design Automation and Test in Europe* (2012), pp. 326–331
4. E. Takeda, Y. Nakagome, H. Kume, S. Asai, New hot-carrier injection and device degradation in submicron MOSFETs. IEEE Proc. I (Solid-State Electron Dev.) **130**(3), 144–150 (1983)
5. T. Ning, C. Osburn, H. Yu, Emission probability of hot electrons from silicon into silicon dioxide. J. Appl. Phys. **48**(1), 286–293 (1977)
6. P.E. Cottrell, R.R. Troutman, T.H. Ning, Hot-electron emission in n-channel IGFET's. IEEE Trans. Electron Dev. **26**(4), 520–533 (1979)
7. B. Grot, S.W. Keckler, O. Mutlu, Preemptive virtual clock: a flexible, efficient, and cost-effective QOS scheme for networks-on-chip, in *EEE/ACM International Symposium on 2009* (2009), pp. 268–279
8. J. Lee, M.C. Ng, K. Asanovic, Globally-synchronized frames for guaranteed quality-of-service in on-chip networks, in *ISCA'08: Proceedings of the 35th Annual International Symposium on Computer Architecture* (2008), pp. 89–100
9. Y. Hoskote, S.R. Vangal, A. Singh, N. Borkar, S. Borkar, A 5-GHz mesh interconnect for a teraflops processor. IEEE Micro **27**(5), 51–61 (2007)
10. D. Ernst, S. Das, S. Lee, D. Blaauw, T.M. Austin, T.N. Mudge, N.S. Kim, K. Flautner, Razor: circuit-level correction of timing errors for low-power operation. IEEE Micro **24**(6), 10–20 (2004)
11. H. Kufluoglu, Mosfet Degradation due to NBTI and HCI and Its Implications for Reliability-Aware VLSI Design, Ph.D. dissertation, Purdue University, West Lafayette, IN (2007)
12. K. Bhardwaj, K. Chakraborty, S. Roy, Towards graceful aging degradation in NoCs through an adaptive routing algorithm, in *DAC Design Automation Conference 2012* (2012), pp. 382–391
13. D.M. Ancajas, K. Chakraborty, S. Roy, Proactive aging management in heterogeneous NoCs through a criticality-driven routing approach, 2013, pp. 1032–1037
14. D.M. Ancajas, J.M. Nickerson, K. Chakraborty, S. Roy, HCI-tolerant NoC router microarchitecture, in *2013 50th ACM/EDAC/IEEE Design Automation Conference (DAC)* (IEEE, New York, 2013), pp. 1–10
15. D.M. Ancajas, K. Chakraborty, S. Roy, J.M. Allred, Tackling QoS-induced aging in exascale systems through agile path selection, in *2014 International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS)* (2014), pp. 1–10
16. R.J. Shridevi, D.M. Ancajas, K. Chakraborty, S. Roy, Tackling voltage emergencies in NoC through timing error resilience, in *2015 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)* (2015), pp. 104–109
17. P. Basu, R.J. Shridevi, K. Chakraborty, S. Roy, Iconoclast: tackling voltage noise in the NoC power supply through flow-control and routing algorithms. IEEE Trans. VLSI Syst. **25**(7), 2035–2044 (2017)
18. R. Das, O. Mutlu, T. Moscibroda, C.R. Das, Application-aware prioritization mechanisms for on-chip networks, in *2009 42nd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)* (2009), pp. 280–291
19. Z. Li, J. Wu, L. Shang, R.P. Dick, Y. Sun, Latency criticality aware on-chip communication, in *2009 Design, Automation & Test in Europe Conference & Exhibition* (2009), pp. 1052–1057
20. B. Datta, W. Burleson, Analysis and mitigation of NBTI-impact on PVT variability in repeated global interconnect performance, in *GLSVLSI '10: Proceedings of the 20th symposium on Great lakes symposium on VLSI* 2010, pp. 341–346
21. J.F. Cantin, J.E. Smith, M.H. Lipasti, A. Moshovos, B. Falsafi, Coarse-grain coherence tracking: RegionScout and region coherence arrays. IEEE Micro **26**(1), 70–79 (2006)
22. D. Fick, N. Liu, Z. Foo, M. Fojtik, J. sun Seo, D. Sylvester, D. Blaauw, In situ delay-slack monitor for high-performance processors using an all-digital self-calibrating 5 ps resolution time-to-digital converter, in *2010 IEEE International Solid-State Circuits Conference - (ISSCC)* (2010), pp. 188–189
23. S. Das, C. Tokunaga, S. Pant, W.-H. Ma, S. Kalaiselvan, K. Lai, D. Bull, D. Blaauw, RazorII: in situ error detection and correction for PVT and SER tolerance. IEEE J. Solid-State Circ. **44**(1), 32–48 (2009)

24. W.J. Dally, B. Towles, *Principles and Practices of Interconnection Networks* (Morgan Kaufmann, San Francisco, CA, 2004)
25. Y. Kim, L.K. John, S. Pant, S. Manne, M.J. Schulte, W.L. Bircher, M.S.S. Govindan, Audit: stress testing the automatic way, in *2012 45th Annual IEEE/ACM International Symposium on Microarchitecture* (2012), pp. 212–223
26. A.K. Kodi, A. Sarathy, A. Louri, J. Wang, Adaptive inter-router links for low-power, area-efficient and reliable Network-on-Chip (NoC) architectures, in *2009 Asia and South Pacific Design Automation Conference* (IEEE, New York, 2009), pp. 1–6
27. D. Zoni, W. Fornaciari, NBTI-aware design of NoC buffers, in *Proceedings of the 2013 Interconnection Network Architecture: On-Chip, Multi-Chip* (2013), pp. 25–28
28. J. Alshraiedeh, A. Kodi, An adaptive routing algorithm to improve lifetime reliability in NoCs architecture, in *2016 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)* (IEEE, New York, 2016), pp. 127–130
29. L. Wang, X. Wang, T. Mak, Dynamic programming-based lifetime aware adaptive routing algorithm for network-on-chip, in *2014 22nd International Conference on Very Large Scale Integration (VLSI-SoC)* (IEEE, New York, 2014), pp. 1–6
30. J. Heißwolf, R. König, J. Becker, A scalable NoC router design providing QoS support using weighted round robin scheduling, in *2012 IEEE 10th International Symposium on Parallel and Distributed Processing with Applications* (IEEE, New York, 2012), pp. 625–632
31. S. Avramenko, S.P. Azad, S. Esposito, B. Niazmand, M. Violante, J. Raik, M. Jenihhin, QoSinNoC: analysis of QoS-aware NoC architectures for mixed-criticality applications, in *2018 IEEE 21st International Symposium on Design and Diagnostics of Electronic Circuits & Systems (DDECS)* (IEEE, New York, 2018), pp. 67–72
32. R. Tamhankar, S. Murali, S. Stergiou, A. Pullini, F. Angiolini, L. Benini, G. De Micheli, Timing-error-tolerant network-on-chip design methodology. IEEE Trans. Comput.-Aided Des. Integr. Circ. Syst. **26**(7), 1297–1310 (2007)
33. D. DiTomaso, T. Boraten, A. Kodi, A. Louri, Dynamic error mitigation in NoCs using intelligent prediction techniques, in *2016 49th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)* (IEEE, New York, 2016), pp. 1–12
34. V.Y. Raparti, S. Pasricha, PARM: power supply noise aware resource management for NoC based multicore systems in the dark silicon era, in *Proceedings of the 55th Annual Design Automation Conference*, 2018, pp. 1–6
35. N. Dahir, T. Mak, F. Xia, A. Yakovlev, Minimizing power supply noise through harmonic mappings in networks-on-chip, in *Proceedings of the Eighth IEEE/ACM/IFIP International Conference on Hardware/software Codesign and System Synthesis* (2012), pp. 113–122