Chapter 4
RoCo: The Row–Column Decoupled Router – A Gracefully Degrading and Energy-Efficient Modular Router Architecture for On-Chip Networks [40]

Network-on-chip architectures are required to not only provide ultra-low latency, but also occupy a small footprint and consume as little energy as possible. Further, reliability is rapidly becoming a major challenge in deep sub-micron technologies due to the increased prominence of permanent faults resulting from accelerated aging effects and manufacturing/testing challenges.

Towards the goal of designing low-latency, energy-efficient and reliable on-chip communication networks, we propose a novel fine-grained modular router architecture. While Chapter 3 focused solely on the router buffers, this chapter will attempt to optimize several of the remaining core micro-architectural components. In particular, the proposed architecture employs decoupled parallel arbiters and uses smaller crossbars for row and column connections to reduce output port contention probabilities as compared to existing designs. Furthermore, the router employs a new switch allocation technique known as "Mirroring Effect" to reduce arbitration depth and increase concurrency. In addition, the modular design permits graceful degradation of the network in the event of permanent faults and also helps to reduce the dynamic power consumption. Our simulation results indicate that in an 8 × 8 mesh network, the proposed architecture reduces packet latency by 4–40% and power consumption by 6–20% as compared to two existing router architectures. Evaluation using a combined performance, energy and fault-tolerance metric indicates that the proposed architecture provides 35–50% overall improvement compared to the two earlier routers.

4.1 Introduction and Motivation

While researchers have proposed performance enhancement techniques [48,73], area-constrained design alternates [19,74], power-efficient and thermal-aware systems [22,70,75], and fault-tolerant mechanisms [35] for NoCs, a systematic design methodology encompassing the interplay of performance, fault-tolerance and energy constraints is yet to evolve. In an attempt to address this issue, this chapter

C. Nicopoulos et al., *Network-on-Chip Architectures: A Holistic Design Exploration*, Lecture Notes in Electrical Engineering 45,
DOI 10.1007/978-90-481-3031-3_4, © Springer Science+Business Media B.V. 2009

presents the design of a modular wormhole-switched router architecture considering the performance, energy, and fault-tolerance issues in a cohesive manner. The salient features of the proposed router that make it distinct compared to other contemporary designs are the following:

(1) For enhancing performance, the virtual channel allocation (VA) and switch allocation (SA) units are cleverly exploited in minimizing the delay due to resource contention.
(2) For energy conservation, the author focuses on use of smaller crossbars, simplified arbiter circuits, and other design tricks such as early ejection and mirrored arbitration.
(3) For enhancing fault-tolerance, the architecture relies on a modular design such that failure of a router component can be tolerated by allowing the switch to operate in a degraded mode. In this context, several techniques are proposed to handle hardware faults in different components of the router.

The proposed Row–Column (RoCo) Decoupled Router enhances performance by reducing the contention probability. This is achieved by splitting the router operation into two distinct and independent modules. Each module is responsible for handling traffic in one dimension (X-dimension or Y-dimension). This decoupling permits the use of smaller and simpler components with reduced logic depth. Each module requires a compact 2×2 crossbar, as opposed to the bigger monolithic crossbar, used in conventional architectures. Furthermore, the proposed router uses a novel switch arbitration scheme, known as the Mirroring Effect. This mechanism requires fewer global arbiters and maximizes crossbar utilization by providing optimal matching between inputs and outputs. Finally, contention in the crossbar is further reduced through the use of a preliminary path-sensitive buffering process, known as Guided Flit Queuing.

The RoCo router possesses inherent fault-tolerant attributes. Its decoupled operation allows for partial functionality in the event of a hard failure. Having two operationally independent modules implies that one module can continue to provide service in one dimension even if the other module is blocked due to a permanent failure. This alleviates contention around the faulty node, which has a profound effect on network latency. Additionally, the proposed architecture employs a hardware recycling mechanism which uses resource sharing to circumvent hard failures in various intra-router components. A comprehensive router fault model is presented and several safeguards are proposed to protect against various types of intra-router faults. These measures induce minimal area, power and latency overheads.

A flit-level, cycle-accurate simulator is used to analyze the performance of the proposed architecture and compare it against a generic 2-stage router and the Path-Sensitive router [73] under a variety of traffic patterns. Moreover, the routers are synthesized in 90 nm technology to extract their power profiles. The power numbers are imported into the simulator for detailed energy analysis. The three router types are used to analyze the performance of an 8×8 mesh network with deterministic, XY–YX and adaptive routing. Simulation results show that the proposed architecture reduces packet latency by 4–40% and power consumption by 6–20% as compared to the two aforementioned router architectures. Evaluation using a combined performance, energy and fault-tolerance metric indicates that our architecture provides a substantially improved combination of high performance, low energy and effective fault-tolerance (almost 50% improvement compared to the generic router and 35% improvement with respect to the Path-Sensitive router).

4.2 Related Work in Partitioned Router Architectures

A comprehensive survey of NoC architectures can be found in [76,77]. In this section, we concentrate predominantly on prior work in partitioned router implementations, where the router architecture is characterized by a modular or hierarchical structure that divides labor to the various components based on an underlying fundamental premise.

Kim et al. [78] proposed a hierarchical switch and cross-point buffering considering the significance of effective allocation for interconnection networks, suitable for parallel computer architectures. The design employs a hierarchical crossbar, in which the sub-switches are interconnected and require global control logic for coordination. Further, a flit may require concatenated crossbar traversal, and use VC buffers at intermediate switching points. The same philosophy is used in this chapter, i.e., *lowering contention*, but with a different design approach for NoCs. The author's approach splits the crossbar into two totally independent and decentralized switches. There is no concatenated switch traversal and no centralized control logic.

The *Path-Sensitive router* supporting routing adaptivity [73] utilizes look-ahead routing in selecting the next route. The router is called Path-Sensitive because – based on the destination address – it has four sets of VCs, called path sets; one set for possible traversal in each of the four quadrants: NE, SE, NW, and SW. Each path set has three groups of VCs to hold flits from possible directions from the previous router. The architecture utilizes a 4×4 decomposed crossbar design with half the connections of a full crossbar. It was shown that the Path-Sensitive router can reduce the average latency compared to a 2-stage router. While this is a nice approach to reduce network latency, it will be shown in this chapter that one can do better by reducing the crossbar size and employing a decoupled design for better concurrency and fault-tolerance. The Partitioned Dimension-Order Router (PDR) [79,80] uses two 3×3 crossbars. However, the operation of the two crossbars is intertwined and the flits should take concatenated switch traversals in order to change dimension. Choi and Pinkston [81] also proposed partitioned crossbar architectures exploring spatial locality and the fact that a packet tends to traverse the network in the same virtual channel. However, these schemes are different compared to the proposed architecture.

4.3 The Proposed Row–Column (RoCo) Decoupled Router

4.3.1 Row–Column Switch

Figure 4.1a illustrates the architecture of a generic 5-port, 2-stage NoC router employing virtual channel flow control and wormhole switching. As already explained in Chapter 2, the five ports correspond to the four cardinal directions and the connection to the local Processing Element (PE). The router consists of six major components: the Routing Computation unit (RC), the Virtual Channel Arbitration (VA), the Switch Allocator (SA), the MUXes and DEMUXes which control the flit flow through the router, the VC buffers, and the crossbar. It employs a pipelined design with speculative path selection to improve performance. Instead of relying on a unified architecture with a monolithic crossbar, the proposed router consists of dual compact crossbars arranged in Row and Column Path Sets. Figure 4.1b depicts the major components of the new 2-stage, pipelined router architecture. The first stage is responsible for look-ahead routing, virtual channel allocation (VA) and speculative switch allocation (SA); all three operations are performed in parallel. The second stage is responsible for crossbar traversal. In this work, the functionality of the router is described with respect to a 2D mesh interconnect.

The router has two sets of crossbars, called Row-Module (East–West) and Column-Module (North–South). The router is divided into two distinct, independent units, each responsible for possible traversal in the corresponding crossbar connections: i.e., in the East–West direction or in the North–South direction. Each port of the crossbar module has a set of three VCs to hold arriving flits from neighboring routers or the local PE. These sets are aptly named Path Sets, since all flits within such a set travel in the same physical direction. In order for an incoming header flit to pass through the DEMUX and be placed into the buffer corresponding to its output path, the header flit should know its route before departing the previous node. To remove the routing process from the router's critical path, the Routing Computation (RC) can be performed one step ahead. By employing this Look-Ahead Routing scheme, the flit is guided to the appropriate buffer by the DEMUX.

Based on the required output port, the header flit requests a valid output VC. The virtual allocation unit, VA, arbitrates between all packets requesting access to the same VCs and decides on winners. Figure 4.2 compares the complexity of the VA unit of a generic 5-port (North, East, South, West, and PE) router and the proposed RoCo router. The use of early ejection allows the RoCo router to eliminate the PE path set (early ejection is analyzed later on in this sub-section). In this comparison, we assume v VCs per input port for both the generic and RoCo architectures. Figure 4.2 compares two cases: one, where the routing function returns a single virtual channel ($R=>v$), and one, where the routing function returns a single physical channel ($R=>p$). Clearly, the RoCo router requires fewer ($4v$ vs $5v$) and smaller ($2v$:1 vs $5v$:1) arbiters in both cases. This attribute significantly reduces the complexity of the arbitration process, since smaller and fewer arbiters imply less contention

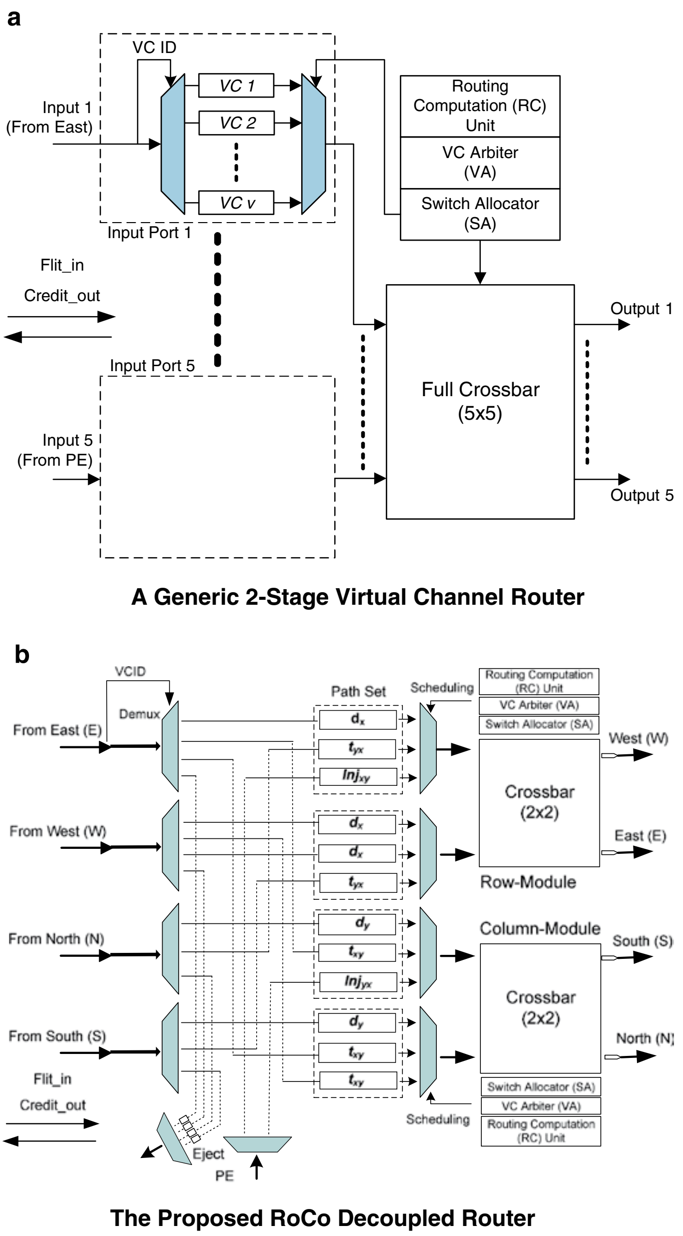

Fig. 4.1 On-chip router architectures: (**a**) a generic 2-stage virtual channel router, (**b**) the proposed RoCo Decoupled Router

and reduced arbitration depth. On the contrary, the increased complexity of the VA in a generic architecture requires multiple iterative arbitrations before satisfying all pending requests [49].

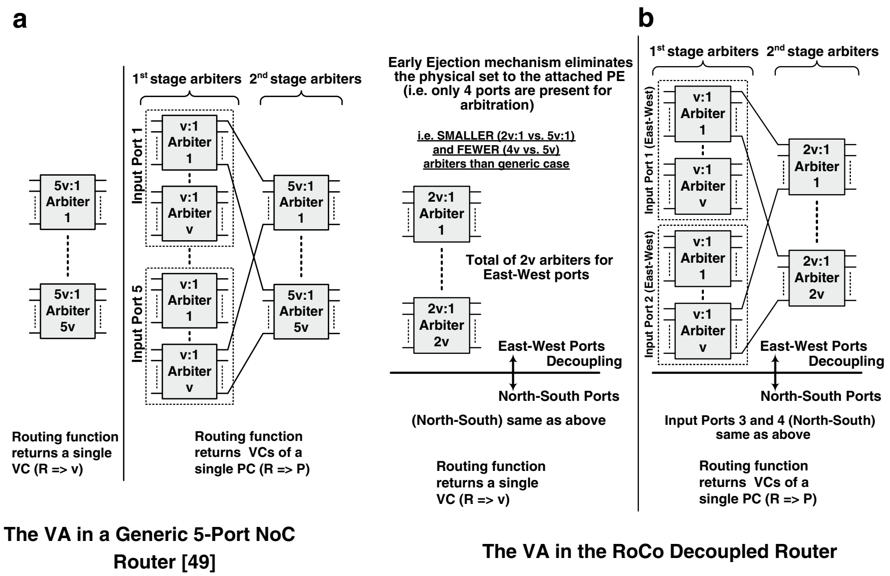
a
5v:1 Arbiter 1
5v:1 Arbiter 5v
Routing function returns a single VC (R => v)
1st stage arbiters
2nd stage arbiters
Input Port 1
Input Port 5
v:1 Arbiter 1
v:1 Arbiter v
5v:1 Arbiter 1
5v:1 Arbiter 5v
Routing function returns VCs of a single PC (R => P)
The VA in a Generic 5-Port NoC Router [49]
b
Early Ejection mechanism eliminates the physical set to the attached PE (i.e. only 4 ports are present for arbitration)
i.e. SMALLER (2v:1 vs. 5v:1) and FEWER (4v vs. 5v) arbiters than generic case
2v:1 Arbiter 1
2v:1 Arbiter 2v
Total of 2v arbiters for East-West ports
East-West Ports
Decoupling
North-South Ports
(North-South) same as above
Routing function returns a single VC (R => v)
1st stage arbiters
2nd stage arbiters
Input Port 1 (East-West)
Input Port 2 (East-West)
v:1 Arbiter 1
v:1 Arbiter v
2v:1 Arbiter 1
2v:1 Arbiter 2v
East-West Ports
Decoupling
North-South Ports
Input Ports 3 and 4 (North-South) same as above
Routing function returns VCs of a single PC (R => P)
The VA in the RoCo Decoupled Router

Fig. 4.2 Virtual channel arbitration (VA) comparison: (**a**) the VA in a generic 5-port NoC router [49], (**b**) the VA in the RoCo Decoupled Router

In the proposed architecture, look-ahead routing decides the valid outgoing channels of packets based on their output paths (Row-Module or Column-Module) and decides whether or not they are continuing along the same dimension. In Fig. 4.1b, VCs marked d_x (d_y) hold flits which continue traversal in their current X (Y) dimension, i.e., East or West (North or South). VCs marked t_{xy} (t_{yx}) hold flits that switch from the X to the Y dimension (Y to X). For example, a flit traversing the network from the east toward the north or the south will arrive at the t_{xy} VC of the first input port in the Column-Module. If the flit is to continue traversal to the west, it is buffered in the d_x VC. That is, d_x and d_y VCs are used for on-going flits along the same dimension, while t_{xy} and t_{yx} are used for changing from the Row-Module to the Column-Module and the other way around. A flit coming from a local PE and destined to the X-dimension, such as the east or the west outputs, is buffered in the Inj_{xy} VC of the Row-Module, while a flit addressed to the Y-dimension is queued into the Inj_{yx} VC of the Column-Module. Depending on the type of routing algorithm used in the network, the number and configuration of the VC buffers changes accordingly. A deadlock free deterministic routing algorithm, such as XY routing, requires a minimum of eight VCs for correct functionality (2 d_x, 2 d_y, 2 t_{xy}, 1 Inj_{xy} and 1 Inj_{yx} for source–destination pairs which lie in the same column). To provide support for deadlock-free XY–YX routing, one additional d_x VC and one additional d_y VC are required. Finally, to provide support for deadlock-free adaptive routing, one more t_{xy} VC and one more t_{yx} VC are needed, making it a total of 12 VCs, as shown in Fig. 4.1b.

Table 4.1 VC buffer configuration for the three routing algorithms

	Row-module		Column-module	
Input port	Port 1	Port 2	Port 1	Port 2
Adaptive	$d_x\ t_{yx}\ Inj_{xy}$	$d_x\ d_x\ t_{yx}$	$d_y\ t_{xy}\ Inj_{yx}$	$d_y\ t_{xy}\ t_{yx}$
XY–YX	$d_x\ t_{yx}\ Inj_{xy}$	$d_x\ d_x\ t_{yx}$	$d_y\ t_{xy}\ Inj_{yx}$	$d_y\ d_y\ t_{xy}$
XY	$d_x\ d_x\ Inj_{xy}$	$d_x\ d_x\ Inj_{xy}$	$d_y\ t_{xy}\ Inj_{yx}$	$d_y\ d_y\ t_{xy}$

These VCs are grouped into four path sets, each containing three VCs. When the router is used with deterministic or XY–YX routing (which can operate with less than 12 VCs), the extra VCs are re-assigned to improve performance by reducing the Head-of-Line (HoL) blocking. For example, XY routing gives rise to asymmetric utilization of the router; HoL in the X-dimension happens more frequently than in the Y-dimension, and the injection channel Inj_{xy} is much more frequently used than Inj_{yx} as a result of the routing scheme. To account for this unbalanced traffic distribution, two additional d_x VCs are assigned to the extra buffers available in the router. Similarly, all 12 VCs present in the router are assigned differently, according to the routing algorithm used. The VC buffer configurations for the three supported routing algorithms (XY, XY–YX, and adaptive) are summarized in Table 4.1.

Upon successful VC allocation and provided a buffer space is available in the downstream router, a flit requests access to the crossbar by undergoing Switch Allocation (SA). The SA arbitrates between all VCs requesting access to the crossbar and grants permission to the winning flits. The winning flits are then able to traverse the crossbar and are forwarded to the respective output links. Switch arbitration works in two stages; stage 1 requires a v-input arbiter for each input port (since there are v VCs per port). Stage 2 arbitrates between the winners from each input port and requires P P-input arbiters, where P is the number of physical ports. The proposed architecture splits the SA module into two smaller modules, each responsible for a small 2×2 crossbar. The reduced number of crossbar ports minimizes the complexity of the SA modules, which function independently from each other. Operation of the SA modules is described in detail in Section 4.3.3.

Deadlock Freedom Adding extra VCs is a technique commonly used to provide deadlock freedom in adaptive routing [82,83]. The two d_x VCs in the second path set of the Row-Module provide a deadlock-free path in the East–West direction during a potential deadlock. The location of the two VCs need not be in the second path set. Interchanging the locations of the VCs in the two path sets of the Row-Module would still yield the same effect. The two t_{xy} VCs in the second path set of the Column-Module are used to ensure deadlock-free routing in case of a chained cyclic dependency. The first t_{xy} VC of the Column-Module is used for turning from the east to the south direction, and the second t_{xy} VC is used for turning from the east to the north direction. Once again, the location of these VCs may be interchanged between the two path sets of the Column-Module without affecting performance.

Early Ejection A flit destined for the local PE does not traverse the crossbar, but, instead, it is ejected immediately upon arrival (hence the "Early Ejection" mechanism). This mechanism utilizes the look-ahead routing information to detect if the incoming flit is destined for the local PE and accordingly ejects it after the DEMUX. This early ejection saves two cycles at the destination node by avoiding switch allocation and switch traversal. Also, it reduces the input load for each crossbar input port. This provides a significant advantage in terms of nearest-neighbor traffic, and can take advantage of NoC mapping which places frequently communicating PEs close to each other [84].

Modular Router and Guided Flit Queuing In our proposed architecture, the input decoders (DEMUXes) undertake a more significant role than in a generic router. In the latter, the input decoders can only distribute incoming flits to the VC buffers of a single port set (see Fig. 4.1a). In the RoCo architecture, however, the input decoders can distribute flits to multiple path sets. This mechanism amounts to a preliminary switching operation, which we call "Guided Flit Queuing", and significantly alleviates contention later on in the crossbar by pre-arranging incoming flits according to their desired output path dimension (X or Y). The area consumed by the router is dominated by the buffers. Therefore, while Guided Flit Queuing increases wiring complexity, the wires have plenty of space to be routed above the buffers in upper layers of the chip, thus imposing minimal overhead. Further, a smaller and simpler crossbar structure reduces wiring complexity.

4.3.2 Blocking Delay

Network latency consists of actual transfer time and blocking delay. The blocking delay is heavily influenced by the switch allocation strategy and the traffic pattern, while the actual transfer time is determined by the floor-plan and the topology of the network. Given that the transfer time is defined by the physical design, we address the other component of network latency, i.e., blocking delay due to contention. Contention is a result of the two arbitration processes occurring within the router: virtual channel allocation and crossbar passage (input port service scheduling and output port allocation). Figure 4.3 shows the comparison of the input contention probabilities in three different architectures (Generic, Path-Sensitive [73] and RoCo) in an 8×8 mesh network with uniform traffic pattern. The results are obtained using our cycle-accurate simulator (described in Section 4.5). In Dimension-Order Routing (DOR – XY routing), the flits of the row input are involved in more severe output conflicts than the column input, because of the nature of the routing algorithm (i.e., X first, Y next). Thus, contention at the row input is higher than at the column input, as shown in Fig. 4.3a, b. Adaptive routing is useful for avoiding local congestion, but it does not reduce the contention probability unless an efficient allocation technique is employed. In fact, adaptive routing may have poor performance with uniform traffic, as explained in [82]. It is evident from Fig. 4.3 that the generic router suffers from high contention probability, which inevitably leads to high Head-of-Line

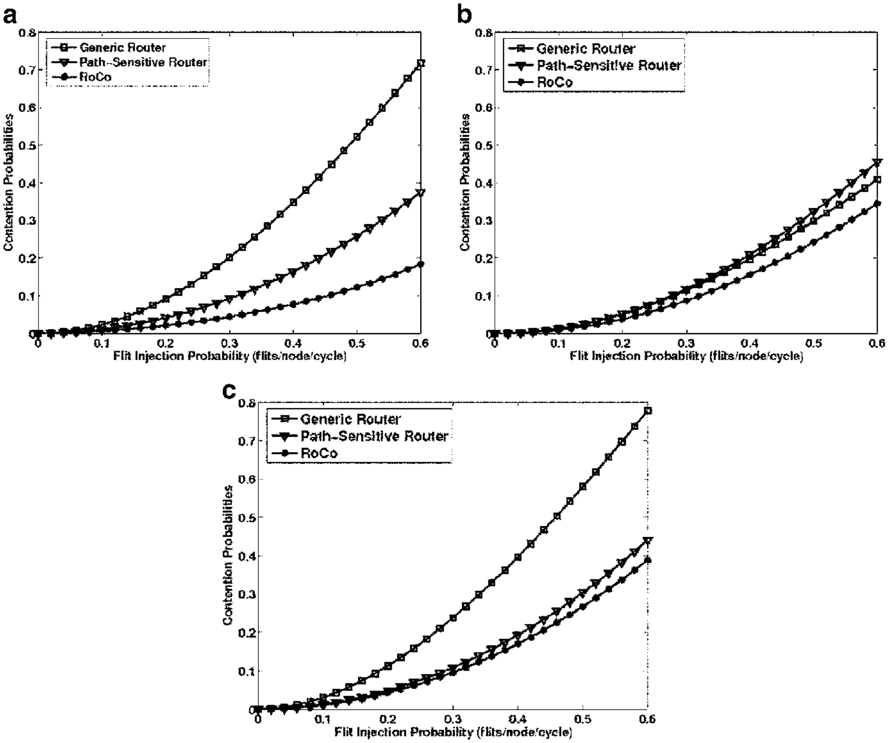

Fig. 4.3 Contention probabilities: (**a**) contention at row input in XY routing, (**b**) contention at column input in XY routing, (**c**) contention in adaptive routing

Table 4.2 Non-blocking probabilities for the three router architectures (N=5)

Route designs	Generic	Path-sensitive	Roco
Non-blocking	$0.043=\left(\dfrac{F(N)}{(N-1)^N}\right)$	$0.125=\left(\dfrac{2}{2^4}\right)$	$0.125=(1-0.5)^2$

(HoL) blocking. The RoCo router has the least contention probability. Furthermore, the RoCo router significantly outperforms the other two architectures in terms of non-blocking probability (i.e., when each output port has one input connection; we call this maximal matching between input and output ports). The non-blocking probabilities for the three router architectures are shown in Table 4.2. Assuming that each input flit has an equal probability $1/(N-1)$ of accessing one of the $(N-1)$ output ports in an $N\times N$ crossbar, the number of cases in which non-blocking maximal matching, $F(N)$, occurs is computed as

$$F(N)=!-\sum_{j=1}^{N}\binom{N}{j}F(N-j)\ \ where\,N>=3, F\ (1)=0\,and\,F(2)=1 \qquad (4.1)$$

In the Path-Sensitive router proposed in [73], arriving flits are grouped in sets depending on their destination quadrant (North–East, North–West, etc.). In this architecture, two inputs from each quadrant path set request one output port. For example, flits in the two quadrants NE and NW may compete for the north output channel. In a similar fashion, two input ports also compete for one output in the RoCo router. However, RoCo uses parallel and independent crossbars, while the Path-Sensitive router has chained dependency between requests. Thus, only 2 cases out of 2^4 matches are non-blocking in the Path-Sensitive router, while 2 cases out of 2^2 matches are non-blocking in each module of the RoCo router. The RoCo router is almost six times more likely to achieve maximal matching than a generic router (25–4.3%), and two times more likely than the Path-Sensitive router (25–12.5%). This implies that the RoCo design is better in terms of providing non-blocking connections.

4.3.3 Concurrency Control for High-Contention Environments

In this section, we introduce the “Mirroring Effect”, a new switching allocation scheme that provides maximal matching in the RoCo router. The Mirroring Effect is a simple algorithm that finds the maximum number of matches between inputs and outputs, customized to the small 2×2 crossbar of each module. The two sets of disjoint pair-wise switch allocators are illustrated in Fig. 4.4. The algorithm is based on the rationale that maximal matching is achieved when the switch allocation results of the two input ports of a single module are mirror images of each other. This realization allows the RoCo implementation to perform global arbitration in only one of the two input ports of each module, and the result is mirrored in the other port. This is illustrated on the right-hand side of Fig. 4.4. For example, if a flit in the top input port is to be forwarded to the West direction, then the bottom port should forward a flit to the East direction to ensure full utilization of the crossbar. Hence, the bottom input port grants access to a flit which wants to continue traversal in the East direction. This scheme constitutes a simple and concurrent global arbitration mechanism compared to the complex hierarchical arbitrations and Parallel Iterative Matching (PIM) [82]. Even though the global switch arbitration decision in the proposed Mirror Allocator is made at the first port, the allocator also gets state information from the bottom port (Fig. 4.4), ensuring that maximal matching is always achieved at each crossbar.

The proposed mechanism requires two arbiters per input port for the first (local) stage of arbitration, as opposed to just one in the generic case. The two arbiters are required to ensure maximal matching by providing the winning requests for both directions (East–West or North–South). However, this small overhead is compensated by the fact that only one arbiter is required per module (because of the Mirroring Effect) in the second (global) arbitration stage (see Fig. 4.4). The mirror arbiter is ideal for a high-throughput switch, because it resolves HoL blocking, eliminates iterative arbitrations, and reduces the inefficiency of local arbitration.

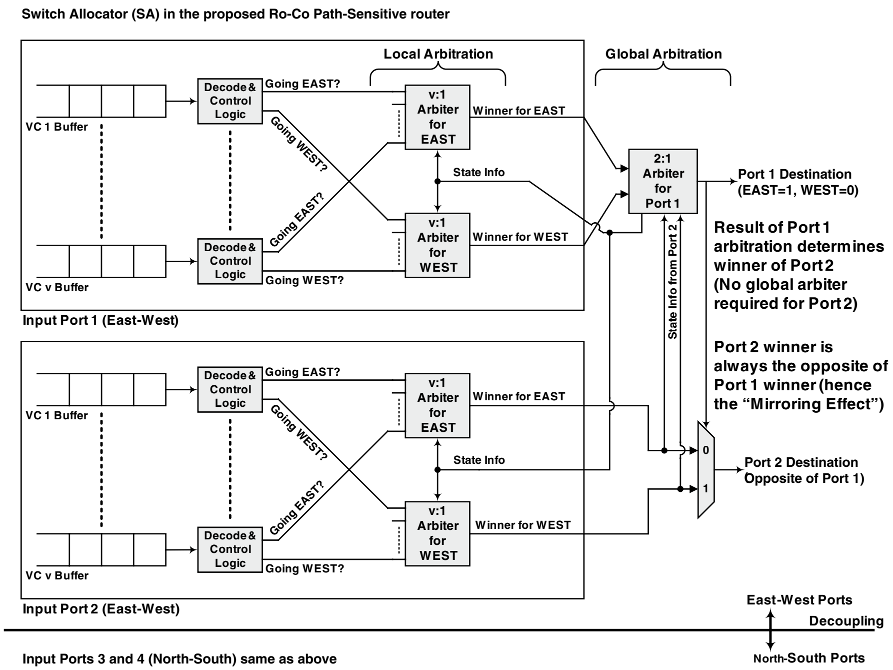
Switch Allocator (SA) in the proposed Ro-Co Path-Sensitive router
Local Arbitration
Global Arbitration
VC 1 Buffer
VC v Buffer
Decode & Control Logic
Going EAST?
Going WEST?
v:1 Arbiter for EAST
v:1 Arbiter for WEST
Winner for EAST
Winner for WEST
State Info
Input Port 1 (East-West)
Input Port 2 (East-West)
2:1 Arbiter for Port 1
Port 1 Destination (EAST=1, WEST=0)
State Info from Port 2
Result of Port 1 arbitration determines winner of Port 2 (No global arbiter required for Port 2)
Port 2 winner is always the opposite of Port 1 winner (hence the "Mirroring Effect")
0
1
Port 2 Destination Opposite of Port 1)
East-West Ports
Decoupling
North-South Ports
Input Ports 3 and 4 (North-South) same as above

Fig. 4.4 The proposed Mirror Allocator

4.3.4 Flexible and Reusable On-Chip Communication

The routing logic, virtual channel arbitration, switch allocation and switching hardware are all partitioned into two separate and independent modules (row and column sets). This decoupling allows for partial operation in case a component within the router malfunctions or suffers a hard failure. In generic router architectures, a hard failure may cause the entire node to be taken off-line, since the operation of the router is unified between all components. In the RoCo router, however, the two disjoint modules function independently. Should a component fail, only the affected module is isolated, with full operation in the remaining module still possible. This would allow the afflicted router to handle network traffic, albeit in limited directions. The fault-tolerance advantages of the RoCo router are analyzed in Section 4.4.

4.4 Fault-Tolerance Through Hardware Recycling

Utilizing the properties of the proposed modular router, a new "Hardware Recycling" mechanism is introduced to ensure fault-tolerant operation of the router. In this section, we explore various possible failure modes within an NoC router, and propose detailed recovery schemes with minimum area and power cost. Our proposed RoCo router architecture possesses some inherent fault-tolerance due to its decoupled design. This additional operational granularity may be utilized to allow replacement of a faulty component by another one, thus allowing partial operation of the router instead of a complete breakdown. The substitution of defective elements by healthy ones elsewhere in the system provides a kind of virtual recycling bin, where functional components can be reused in other parts of the implementation should the need arise. Our proposed scheme avoids the more traditional approach in fault-tolerance, which resorts to replication of resources. Silicon real-estate and energy are at a premium in on-chip applications, thus necessitating the efficient re-use of existing resources.

The six major components of the router – the Routing Computation Unit (RC), the Virtual Channel Arbiter (VA), the Switch Allocator (SA), the MUXes and DEMUXes which control the flit flow through the virtual channel buffers, the VC buffers, and the crossbar – are susceptible to different types of permanent faults. These components can be classified into two categories, based on their operational regime: (a) per-packet components, and (b) per-flit components. Per-packet components (i.e., the RC and VA) are only used to process the header flit of a new incoming packet. The subsequent flits simply follow the wormhole created by the header flit. Per-flit components (i.e., the remaining components) are used to process every single flit passing through the router. Clearly, since the per-packet based components are driven only by the header flit, their utilizations are relatively low compared to the flit-by-flit operation of per-flit components; the latter are fully utilized in non-blocked operation. Thus, packet-based resources can be shared during their unloaded periods.

We further sub-divide the fundamental router components into two classes: message-centric and router-centric. A message-centric component requires a single individual packet as its input, and does not exhibit any interdependencies with other incoming messages. The Routing Computation Unit (RC) and the virtual channel buffers are such examples; they operate on a single message (i.e., packet) and their operation does not require state information from other components within the router. On the other hand, router-centric components require inputs from several pending messages in order to execute their function. The VA and SA are such examples; they arbitrate between all messages requesting passage through the router, and their functionality requires state information from the buffers and adjacent routers.

Finally, it is important to note that the operation of the router consists of a critical pathway and non-critical control logic. The datapath of the router (i.e., guided passage of a flit and switch traversal) constitutes the critical pathway; it consists of buffers, decoders, multiplexers and the crossbar. It should be noted that even though the VC buffers lie in the critical datapath, they may or may not be classified as critical, depending on the presence or not of a bypass path. If bypass paths are employed in the buffers for performance optimization, then the VC buffers can be classified as non-critical because of the redundancy supplied by the extra path as explained later on in the section. Otherwise, the buffers are classified as critical. The operation of the control logic – comprised mostly of the arbiters of the VA and SA – lies in a non-critical pathway. Table 4.3 illustrates the fault classifications of the router components.

Table 4.3 Component fault classification

	Per-flit operation		Per-packet operation	
Fault type	Critical pathways	Non-critical pathways	Critical pathways	Non-critical pathways
Message centric	MUX/DEMUX buffer (w/o bypass path)	Buffer (with bypass path)	–	RC
Router centric	Crossbar	SA	–	VA

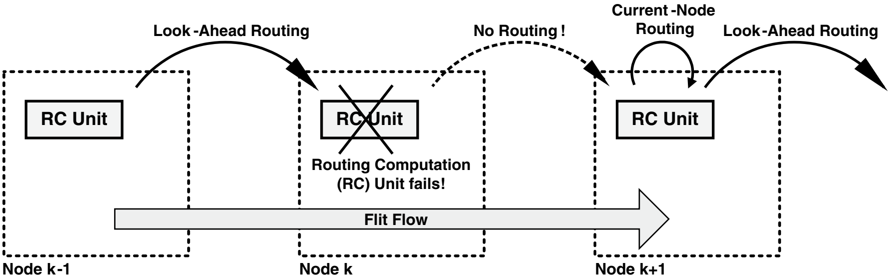

Fig. 4.5 Double Routing mechanism in the event of RC unit failure

Each router node is assumed to be able to detect a faulty component through the use of simple control signals. The novelty in our approach lies in the reaction of the router to a hard failure. If a faulty component belongs to a message-centric and non-critical region, the failure can be bypassed instead of resorting to blocking of the whole router module (Row-Module or Column-Module). We can still partially use the router module with the faulty component. If the faulty block lies on the critical pathway, or if it is a router-centric component, the permanent failure cannot be bypassed. In this case, the module is isolated and the router remains partially operational through the use of the other parallel module in our proposed scheme. Operational state is tracked by neighboring routers through the use of simple handshaking signals. We assumed permanent failures to be handled statically. Upon failure, any fragmented packets are simply discarded. Most of the fault-tolerant schemes proposed in this work can be retrofitted to existing router designs. However, they are particularly amenable to the RoCo router because of its decoupled nature that allows for graceful degradation. The recovery schemes proposed for each component failure are outlined below:

Routing Computation Unit (RC) Failure A hard fault in the routing unit logic could cause all flits to be forwarded in the same direction or, in a more severe case, completely halt the generation of routing signals. The misdirection will not cause any data corruption, but it could lead to deadlock in deterministic routing algorithms. As soon as a failure in the RC unit is detected, it is broadcast to the adjacent routers. After knowing the failure status of the RC unit, the adjacent nodes can now substitute for the faulty RC unit by performing double routing, as shown in Fig. 4.5. Neighboring nodes sending flits to the faulty router need not worry, because their look-ahead routing will ensure that data arriving at the faulty node has already been taken care of. The problem affects the nodes receiving data from the faulty router; flits arriving at those nodes have not undergone look-ahead routing, because of the faulty RC unit in the previous router.

Therefore, nodes receiving flits from the faulty router must first conduct Current-Node Routing on those flits and then proceed to Look-Ahead Routing. The overhead involved is minimal and comes only from the few additional control signals; no additional resources are required.

Buffer Failure In a typical wormhole router, when a flit enters an input port, it is written to an input buffer queue. Bypassing the input buffer when the buffer is empty is a common optimization for performance; the flit heads straight to switch arbitration, and if it succeeds, the flit gets sent directly to the crossbar switch, circumventing the input buffers. This bypass path connecting the router input port with the crossbar input port can also be utilized in the event of buffer failure within a node. Virtual buffer management and switch allocation can still be performed in the current node, but buffer storage is offloaded to the previous node.

As soon as a flit stored in the previous node wins the switch arbitration in the current node, it can use the bypass path to circumvent the faulty buffer and proceed to the crossbar of the current node. In essence, data is physically stored in another router, but virtually queued and arbitrated in a different node through control signals between neighboring routers, as shown in Fig. 4.6. Under the Virtual Queuing mechanism, each buffer is not tied to a single VA arbiter in adjacent routers. Thus, a failure in a VA arbiter does not leave a particular buffer in a deadlock mode. There is a small latency penalty involving the round-trip delays of the handshaking signals, but it does avert the complete isolation of the faulty node. In terms of area cost, Virtual Queuing incurs minimal overhead, since no additional resource is required.

Virtual Channel Arbiter (VA) Failure Hard faults in this router-centric component are hardly recoverable by simply sharing of router resources. The operation of the VA cannot be offloaded to surrounding nodes, since its operation requires state information from several sources and it exhibits inter-dependencies with other router

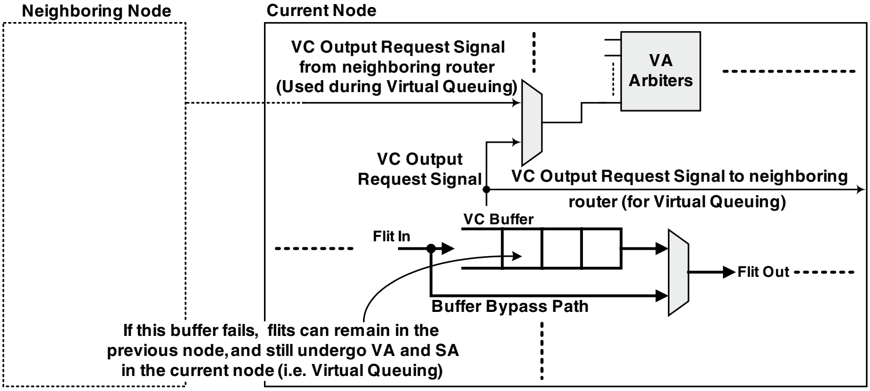
Neighboring Node
Current Node
VC Output Request Signal from neighboring router (Used during Virtual Queuing)
VA Arbiters
VC Output Request Signal
VC Output Request Signal to neighboring router (for Virtual Queuing)
VC Buffer
Flit In
Flit Out
Buffer Bypass Path
If this buffer fails, flits can remain in the previous node, and still undergo VA and SA in the current node (i.e. Virtual Queuing)

Fig. 4.6 The Virtual Queuing mechanism

elements and downstream nodes. Offloading such operations would require excessive transfer of state information. Faults in the VA can be bypassed only through resource replication, which is costly in terms of area and power overhead. The other option is to offload the arbitrations to the Switch Arbiter hardware, which contains identical arbiter modules. Nevertheless, this is infeasible because the SA is a per-flit component, meaning that it operates on all flits on a cycle-by-cycle basis. Since it is fully utilized, its operation cannot be preempted. The only choice, therefore, is to disable the whole router module in the event of a hard failure in the VA. However, whereas in generic architectures that would mean complete isolation of the entire node, in the proposed architecture only one of the two independent modules needs to be disabled.

Switch Allocator (SA) Failure Despite being a router-centric component, the SA can still be saved in the event of a hard failure in one of its components. Its operation cannot be transferred to neighboring routers because of the excessive transfer of state information required by such an endeavor. The solution proposed is much simpler and relies on the fact that the SA uses identical hardware with the VA, which is a per-packet component. Per-packet implies lower utilization, as explained in the beginning of this sub-section. This lends itself nicely to sharing of resources. By including a small number of compact 2-to-1 multiplexers at the input of some of the VA's arbiters, the SA can offload its operation to the VA, as shown in Fig. 4.7. Since the VA is operational only for header flit processing, its arbiters

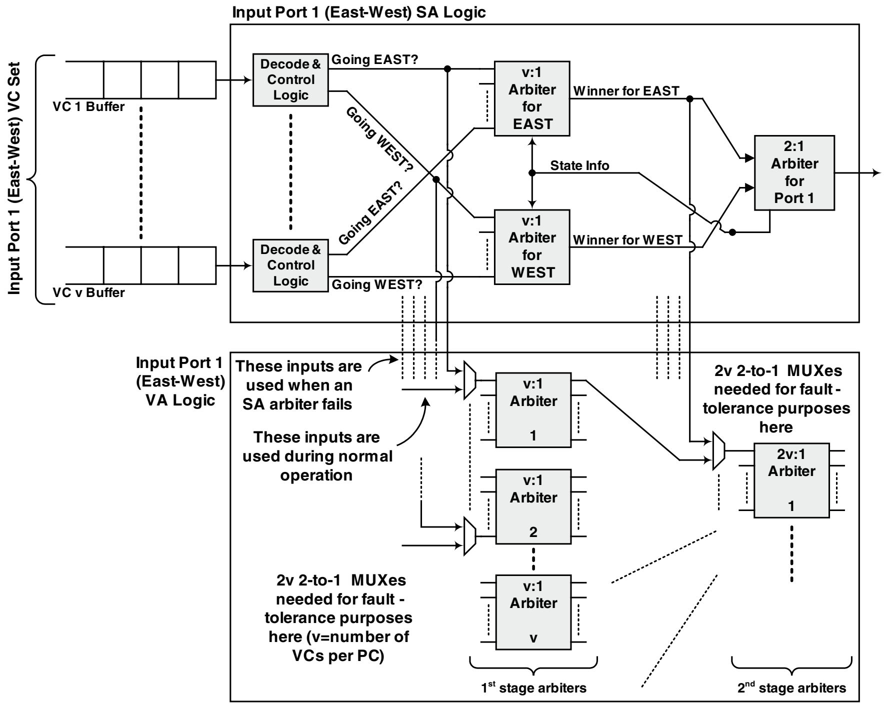
Input Port 1 (East-West) SA Logic
Input Port 1 (East-West) VC Set
VC 1 Buffer
VC v Buffer
Decode & Control Logic
Decode & Control Logic
Going EAST?
Going WEST?
Going EAST?
Going WEST?
v:1 Arbiter for EAST
v:1 Arbiter for WEST
Winner for EAST
Winner for WEST
State Info
2:1 Arbiter for Port 1
Input Port 1 (East-West) VA Logic
These inputs are used when an SA arbiter fails
These inputs are used during normal operation
2v 2-to-1 MUXes needed for fault - tolerance purposes here (v=number of VCs per PC)
v:1 Arbiter 1
v:1 Arbiter 2
v:1 Arbiter v
1st stage arbiters
2v 2-to-1 MUXes needed for fault - tolerance purposes here
2v:1 Arbiter 1
2nd stage arbiters

Fig. 4.7 Switch allocator fault-tolerance through resource sharing

can be used by the SA when they are idle. Performance, of course, is degraded because of the sharing of resources, but it is still a preferable alternative to the complete shutdown of the module (Row-Module or Column-Module). The area and power overhead imposed by the MUXes is minimal.

Crossbar and MUX/DEMUX Faults In the proposed RoCo router architecture, a decoder (DEMUX) is used to guide a flit into a group of path-sensitive queues, and a multiplexer (MUX) is used to direct a winning flit to the crossbar input. Therefore, the MUXes and DEMUXes all lie on the critical pathway of the router. A hard failure in one of these critical components can severely hamper the datapath progression. Once again, bypassing the datapath would imply replication of resources, which is not desirable. Hence, if any of these modules fails, the corresponding router module is blocked, while the other healthy module keeps operating.

4.5 Performance Evaluation

In this section, simulation-based performance evaluation of the RoCo architecture, a generic router architecture and the Path-Sensitive architecture of [73] is presented, in terms of network latency, energy consumption and fault-tolerance under various traffic patterns. The experimental methodology is described, and the procedure followed in the evaluation of these architectures is detailed.

4.5.1 Simulation Platform

A cycle-accurate NoC simulator was developed in order to conduct a detailed evaluation of the router architectures. The simulator operates at the granularity of individual architectural components, accurately emulating the major hardware components. The simulation test-bench models both the routers and the interconnection links, conforming to the implementation of various NoC architectures.

The simulator is fully parameterizable, allowing the user to specify parameters such as network size, topology, switching mechanism, routing algorithm, number of VCs per PC, number of PCs, buffer depth, PE injection rate, injection traffic-type, flit size, and number of flits per packet. The simulator models each individual component within the router architecture, allowing for detailed analysis of component utilizations and flit flow through the network. The activity factor of each component is used for analyzing power consumption within the network. We assume that link propagation happens within a single clock cycle. In addition to the network-specific parameters, our simulator accepts hardware parameters such as power consumption (dynamic and leakage) for each component and overall clock frequency. These parameters are extracted from hardware synthesis tools and back-annotated into the simulator for power profile analysis of the entire on-chip network.

4.5.2 Energy Model

The proposed RoCo router architecture, the Path-Sensitive router of [73] and a generic 2-stage 5-port router architecture were implemented in structural Register-Transfer Level (RTL) Verilog and then synthesized in Synopsys Design Compiler using a TSMC 90 nm standard cell library. The resulting designs both operate at a supply voltage of 1 V and a clock speed of 500 MHz. Both dynamic and leakage power estimates were extracted from the synthesized router implementation, assuming a 50% switching activity. These power numbers were then imported into our cycle-accurate network simulator for power analysis.

4.5.3 A Performance, Energy, and Fault-Tolerance (PEF) Metric

Traditional performance metrics used in NoC analysis, such as the Energy-Delay Product (EDP) and Power-Delay Product (PDP), focus on the two fundamental notions of latency (i.e., performance) and energy/power consumption. These metrics, however, do not capture the importance of reliability and its relation to both performance and power. Given that reliability is becoming a major concern in deep sub-micron technologies, it is imperative that evaluation of NoCs accounts for such issues. To address this need, we propose a composite metric which unifies all the three components: latency, energy, and fault-tolerance. Before introducing the new metric, we define three related terms.

Network Latency This is defined as the average number of cycles taken for end-to-end packet traversal, i.e., from a source to a destination.

Energy Consumption per Packet This is divided into two components: dynamic and leakage energy consumption. Both are defined as the total dynamic (or leakage) energy consumed in the network fabric over a time period divided by the total number of packets delivered during that period. Leakage power captures the effect of blocking delay, which translates into buffer static energy consumption. Dynamic power captures the effect of high contention within the router, which increases energy consumption due to excessive iterative operation of the SA and the VA units.

Packet Completion Probability This is defined as the number of received messages divided by the total number of injected messages into the on-chip network.

The inter-dependence between speed, power and fault-tolerance highlights the importance of a metric which can identify the best tradeoffs between these three competing traits. Hence, we introduce the Performance, Energy and Fault-tolerance (PEF) metric, as a comprehensive parameter that reflects the correlation between the three desired design goals. We define PEF as

$$\begin{aligned} PEF &= \frac{(Average\ \ Latency)\times(Energy-per-Packet)}{Packet\ \ Completion\ \ Probability} \\ &= \frac{Energy-Delay-Product}{Packet\ \ Completion\ \ Probability} \end{aligned} \tag{4.2}$$

In a fault-free network, *Packet Completion Probability* = 1; thus, PEF becomes equal to EDP. Hence, PEF integrates reliability into EDP, thus providing a more complete evaluation metric.

4.5.4 Performance Results

The performance of the proposed RoCo router was analyzed and compared to two other existing router architectures (generic router, Path-Sensitive router of [73]) using the cycle-accurate simulator. All architectures were evaluated using an 8 × 8 2D mesh network. In the generic router architecture, three VCs per port were assumed, with a 4-flit deep buffer per VC; for a 5-port router, this configuration gives a total buffer capacity of 60 flits per router. To ensure fairness, since both the proposed RoCo architecture and the Path-Sensitive router have four ports instead of five, three VCs per port were assumed in both implementations, each with a 5-flit deep buffer; this gives a total buffer capacity of 60 flits per router, similar to the generic case. Each simulation consists of two phases: a warm-up phase of 20,000 packet injections, followed by the main phase which injects 1,000,000 additional packets. Each packet consists of four 128-bit flits. Under normal conditions, the simulation terminates when all packets are received at the destination nodes. In faulty environments, the simulation terminates after a long period of inactivity has elapsed (twice the time required to complete the simulation of a fault-free network).

Several experiments were conducted to evaluate the performance of all architectures under various traffic patterns and three different routing algorithms. The experiments employed uniform and transpose [82] traffic, and two synthesized workload traces, self-similar web traffic [85] and MPEG-2 video multimedia traces [86], in three different routing algorithms: DOR (XY routing), oblivious XY–YX routing, and minimal adaptive routing schemes. The results for multimedia traffic are not included here due to space constraints. Average network latency and power consumption were recorded for all experiments. Furthermore, several experiments were conducted to evaluate performance in faulty environments. A number of router faults (both Message-Centric and Router-Centric, as explained in Section 4.4) were randomly injected into the network infrastructure and the packet completion probability was analyzed. The traffic injection rate in these faulty networks was 30%. The latency, energy and fault-tolerance results were subsequently integrated into the PEF metric of Section 4.5.3 to reflect the combined measure.

The latency results of all three architectures for various traffic patterns are illustrated in Fig. 4.8 through 4.10. Clearly, the proposed RoCo router outperforms both the generic and Path-Sensitive routers in all traffic patterns and routing algorithms. With deterministic routing, the RoCo router reduces average latency by up to 35% compared to the generic router and by about 7% compared to the Path-Sensitive router. With XY–YX routing, these numbers become 38% and 10%, respectively. Finally, in adaptive routing, latency reduces by up to 40% compared to the generic router, and about 4% compared to the Path-Sensitive router.

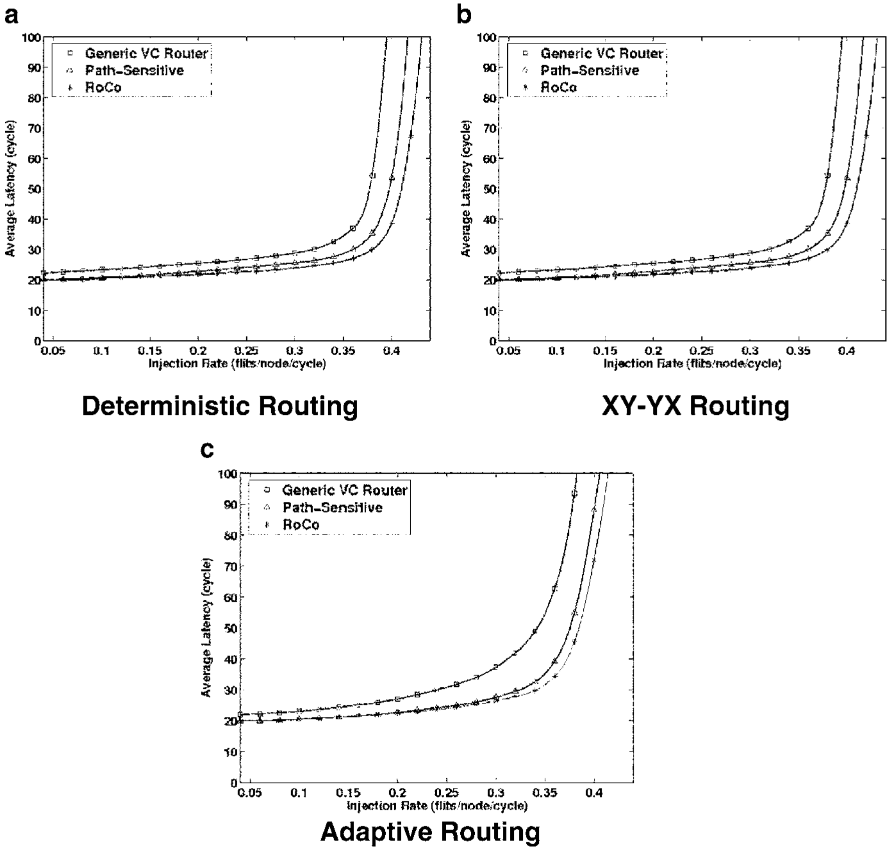

Fig. 4.8 Uniform random traffic: (**a**) deterministic routing, (**b**) XY–YX routing, (**c**) adaptive routing

The decoupling of the architecture into two distinct and functionally independent modules significantly reduces contention probability within the router. This effect manifests itself in lower average latency within the network. Furthermore, the use of the novel Mirroring Effect in switch arbitration increases crossbar utilization and reduces blocking.

Figure 4.13 compares the energy efficiency of the three different router architectures at 30% injection rate. The energy per packet is about 20% lower in the RoCo router, as compared to the generic router architecture, and about 6% lower compared to the Path-Sensitive router. This is a consequence of the simpler crossbars, smaller VA and SA units, and shorter logic depth. Therefore, the benefits afforded by the RoCo router are two-fold: reduced average network latency and lower energy consumed per packet. This is a testament to the fact that a more streamlined architecture can benefit both performance and power consumption.

Figures 4.11 and 4.12 illustrate the packet completion probabilities of the three router architectures when operating in faulty environments with 1, 2 and 4 random

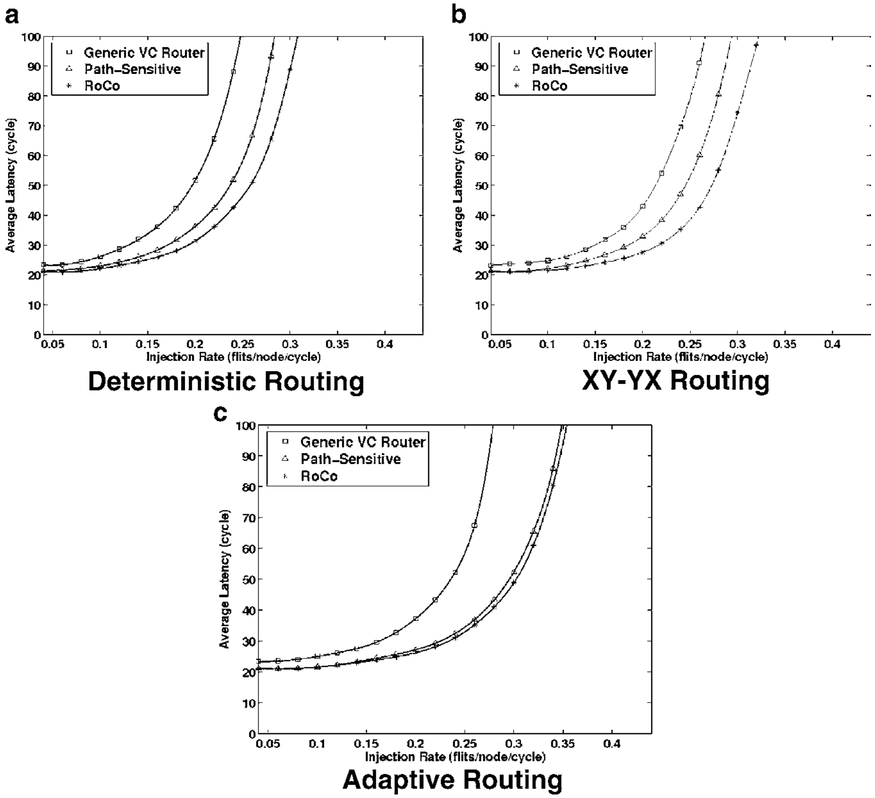

Fig. 4.9 Self-similar traffic: (**a**) deterministic routing, (**b**) XY–YX routing, (**c**) adaptive routing

network faults. Figure 4.11 concentrates on Router-Centric faults. These are critical faults, which cause the entire node to be blocked in the generic and Path-Sensitive cases. In the RoCo architecture, however, such faults only cause one of the two modules (Row-Module or Column-Module) to be blocked, thus, allowing for partial operation of the faulty router. Completion probability is consistently higher in the RoCo router in Fig. 4.11. As the number of faults increases from 1 to 4, the advantage of the proposed router becomes more obvious. The RoCo router provides up to 70% improvement in packet completion probability for different fault patterns with deterministic routing. The improvement drops to about 7% when adaptive routing is used. In both XY–YX and adaptive routing, the results are close because the routing algorithms provide alternate paths for all three architectures. However, this metric alone does not reflect the fact that even though completion probability is high in the generic and Path-Sensitive cases, the latency penalty incurred by excessive congestion around the faulty nodes is very high. This result will be captured later on in the PEF metric (Fig. 4.13).

Figure 4.12 focuses on Message-Centric faults, which are not critical. In the generic and Path-Sensitive routers, such faults would still cause the entire node to

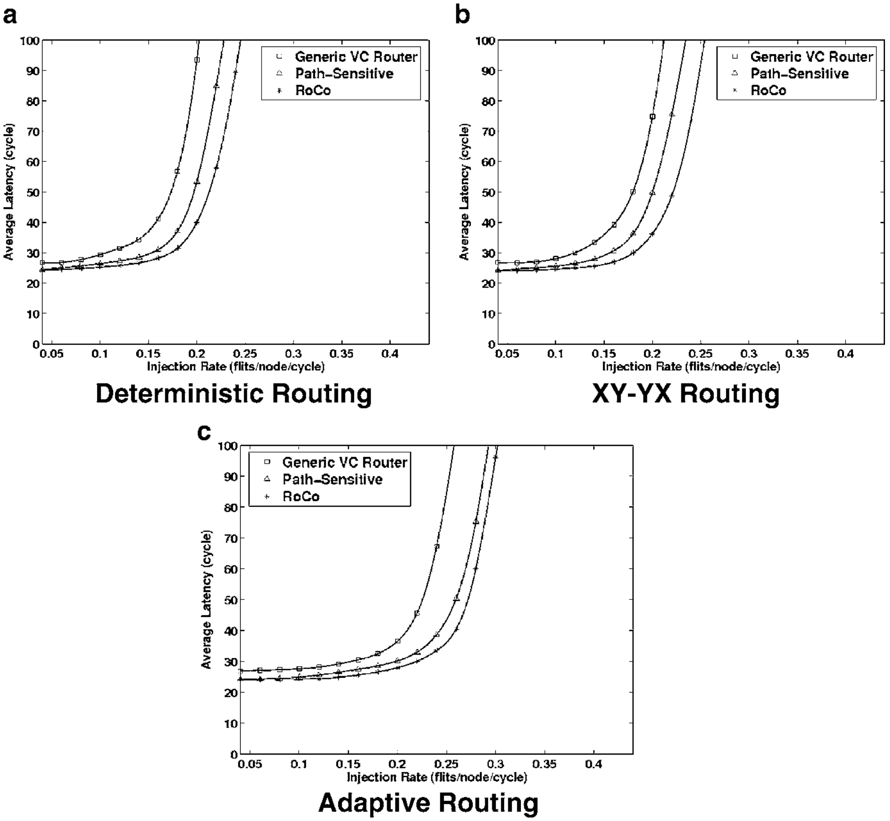

Fig. 4.10 Transpose traffic: (**a**) deterministic routing, (**b**) XY–YX routing, (**c**) adaptive routing

be blocked. However, in the RoCo router, such faults are remedied by the Recycling Mechanism of Section 4.4, which bypasses the faults through resource sharing. The oblivious routing schemes, i.e., deterministic and XY–YX, suffer more in the presence of faults, because of their rigid routing policies. These results indicate that the proposed Recycling Mechanism improves completion probability considerably without any significant router area overhead. Furthermore, even during critical Router-Centric faults, partial operation of the router can still serve network traffic in one dimension, thus alleviating congestion around the faulty node. This is achieved without any additional overhead. The results indicate that the RoCo router can achieve packet completion probabilities in oblivious routing that are close to those of adaptive routing schemes. This is of profound importance, since it indicates that the RoCo router provides uniform fault-tolerance under all routing algorithms. Through the recycling of faulty components and resource sharing, our proposed architecture degrades gracefully in faulty environments.

Figure 4.14 shows the combined measure (PEF) results for the three router architectures. The bars use the scale on the left-hand axis, while the curves use the scale

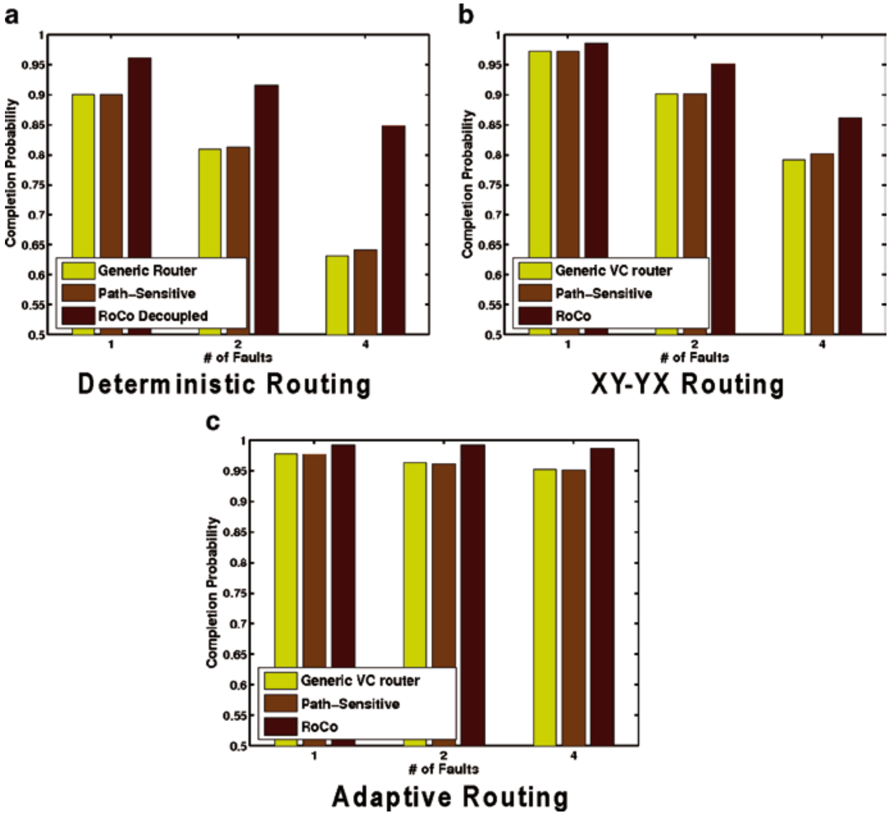

Fig. 4.11 Packet completion probabilities under various faults within router-centric and critical pathway components: (**a**) deterministic routing, (**b**) XY–YX routing, (**c**) adaptive routing

on the right-hand axis. This metric can successfully capture the subtle fact that despite high completion probabilities with adaptive routing, the generic and Path-Sensitive routers suffer from high latency due to congestion created around the faulty nodes. The RoCo router, on the other hand, has significantly lower latency numbers due to graceful degradation and the novel hardware recycling mechanism. Taking into consideration performance, energy consumption, and fault-tolerance in the integrated PEF metric, the RoCo router turns out to be the clear winner compared to the other two architectures. It provides almost 50% improvement compared to the generic router and 35% improvement compared to the Path-Sensitive router.

4.6 Chapter Summary

In this chapter, the author has presented a new router architecture, called Row–Column Decoupled Router, suitable for on-chip interconnects. The uniqueness of the proposed router is that it considers the three desirable objective functions:

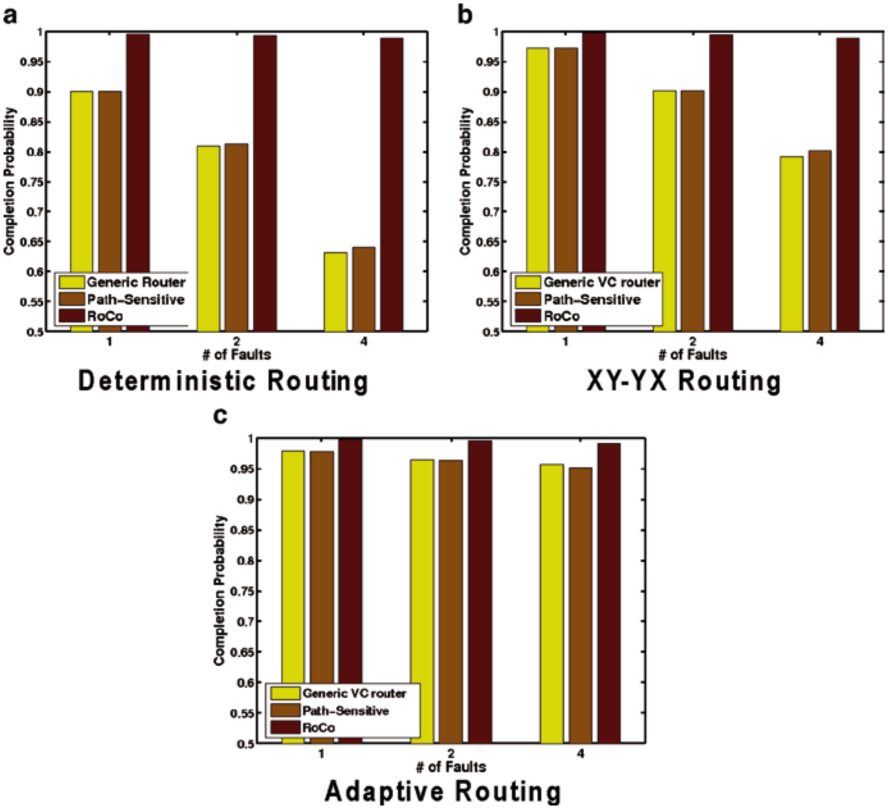

Fig. 4.12 Packet completion probabilities under various faults within message-centric and non-critical pathway components: (**a**) deterministic routing, (**b**) XY–YX routing, (**c**) adaptive routing

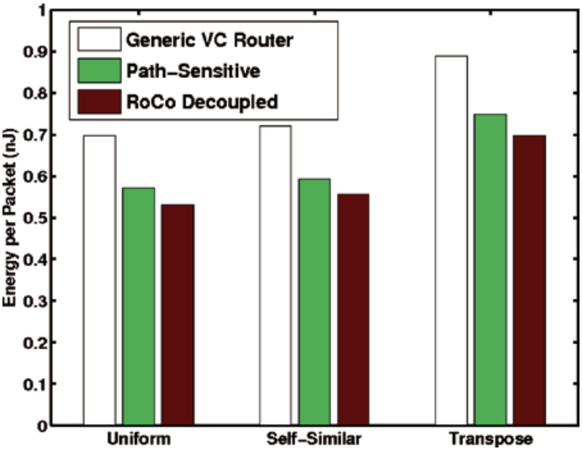

Fig. 4.13 Energy per packet

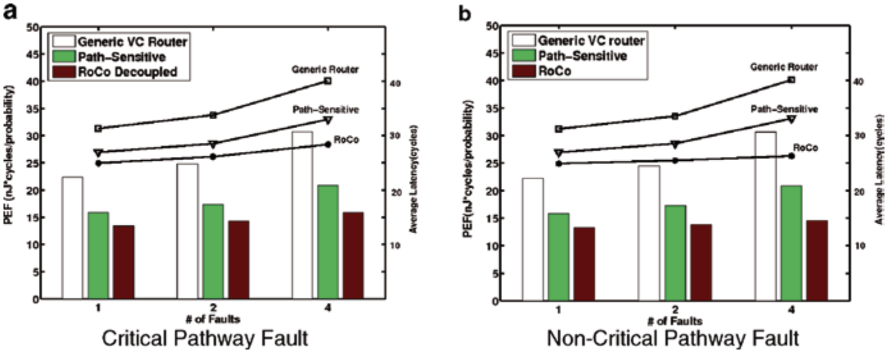

Fig. 4.14 Performance-energy-fault (PEF) product: (**a**) critical pathway fault, (**b**) non-critical pathway fault

performance, energy and fault-tolerance, in exploring the design space. The proposed 2-stage wormhole-switched RoCo router has a number of features that make it distinct compared to the earlier designs. First, it uses two smaller 2 × 2 crossbars instead of a larger 5 × 5 crossbar that is traditionally used for 2D mesh networks. Second, it uses a path-sensitive buffering scheme, where, the virtual channels are divided into four sets to support dedicated row and column routing in the two crossbars. These two features along with early ejection, mirrored allocation, look-ahead routing and speculative path selection help in reducing the contention. Third, unlike most earlier designs, it has been shown how deterministic (XY) routing, XY–YX routing and adaptive routing can be supported in this architecture. Fourth, because of the modular design, it has been shown how different types of faults such as VA, SA, and crossbar failures can be handled with graceful degradation, thereby providing better fault-tolerance compared to earlier designs. In addition, while all prior NoC studies have analyzed at best two of the three parameters, such as energy-delay product, this study introduces a comprehensive parameter, called PEF, for analyzing the performance, energy and fault-tolerance attributes of NoC architectures.

A flit-level, cycle-accurate simulator along with a detailed energy model for 90 nm synthesis were used to analyze the three objective functions using a variety of traffic patterns. Performance analysis with an 8 × 8 mesh network shows that the proposed router can reduce the average network latency up to 40% compared to a generic 2-stage router and by 10% compared to the Path-Sensitive router. In terms of energy consumption per packet, the proposed RoCo design outperformed the 2-stage router and Path-Sensitive router by 20% and 6%, respectively. The packet completion probability is improved by about 70% with deterministic routing. Evaluation with the composite performance, energy and fault-tolerance parameter (PEF) indicates that the RoCo architecture provides 50% and 35% better results compared to the generic and Path-Sensitive models, respectively.