Interconnection Alternatives for Hierarchical Monitoring Communication in Parallel SoCs

Abstract

Interconnection architectures for hierarchical monitoring communication in parallel SoC (System-on-Chip) platforms are explored. Hierarchical agent monitoring design paradigm is an efficient and scalable approach for the design of parallel embedded systems. Between distributed agents on different levels, monitoring communication is required to exchange information, which forms a prioritized traffic class over data traffic. The paper explains the common monitoring operations in SoCs, and categorizes them into different types of functionality and various granularities. Requirements for on-chip interconnections to support the monitoring communication are outlined. Baseline architecture with best-effort service, TDMA (time division multiple access) and two types of physically separate interconnections are discussed and compared, both theoretically and quantitatively on a NoC (Network-on-chip)-based platform. The simulation uses power estimation of 65nm technology and NoC microbenchmarks as traffic traces. The evaluation points out the benefits and issues of each interconnection alternative. In particular, hierarchical monitoring networks are the most suitable alternative, which decouple the monitoring communication from data traffic, provide the highest energy efficiency with simple switching, and enable flexible reconfiguration to tradeoff power and performance.

Key words: Hierarchical Monitoring Services; Network-on-Chip; Interconnection Architectures; Quality-of-Service

1 Introduction

Constant transistor scaling enables the integration and implementation of increasingly complex functionalities onto a single chip. Recently, a 167-processor computational platform was prototyped [1]. Along with the system integration, several major issues challenge the design of on-chip parallel systems. Deep submicron effects (DSM), brought by the feature size shrinking of transistors, will be more profound with sub-65nm technology, including crosstalk effects, capacitance coupling and wire inductance. Process, thermal and voltage (PVT) variations, brought by uncertainties in fabrication and run-time operations, introduce unpredictable performance and errors at every architectural level [2]. Implementation constraints continue to complicate the design process. While silicon area still requires optimization, power consumption becomes more critical along with other physical consideration for instance thermal hotspot. Despite these technological and architectural difficulties, the design time is always expected to be shortened due to the time-to-market pressure.

Facing these design challenges, dynamic monitoring services have been acknowledged as an effective paradigm. It broadly includes many types of operations which dynamically observe, configure and optimize components at various architectural levels. For on-chip parallel systems, a variety of dynamic monitoring operations are needed to deal with varying workloads and component status. These monitoring operations, performed on different architecture levels, require specific handling intervals based on their timing constraints. Common monitoring services will be analyzed in Section 2.1.

Hierarchical agent monitored system, a novel approach to design various monitoring services on a parallel system, was initially proposed in our previous work [3] and further explained in [4]. Following this design approach, monitoring operations are partitioned, based on their granularities, onto hierarchically organized and distributedly located monitors. The design concentration of monitoring services is motivated by the orthogonalization of design concerns [5], which reduces the design complexity and improves design reuse and portability in similar platforms. The functional partition of monitoring operations provides scalability for a platform with any number of parallel components.

Interconnection network, the underlying support for monitoring communication, needs to fulfill the requirements of monitoring operations. Guided by the concept of QoS (Quality of Service), interconnection alternatives with best-effort or guaranteed transmission have been explored in previous works [6, 7]. A few of them have discussed the necessity of providing QoS to certain monitoring operations [7]. However, most previous works did not differentiate the large variety of monitoring services by their specific requirements. Driven by current technology trend, the design of interconnection networks including the realization of QoS needs to consider the emerging considerations. Power and energy efficiency, in particular, has become one of the most important concerns [8, 9], since the power density is increasing much faster than the battery and cooling capacity.

This paper analyzes the features of various types of monitoring services in the hierarchical agent monitored platform, and outlines the interconnection requirements to support these features. Several interconnection alternatives, including best-effort architecture, TDMA-based approach and physically separate networks are discussed and experimented with diverse settings. The simulation is performed with $65nm$ power and area parameters on an $8*8$ on-chip network platform. From these theoretical and quantitative analysis, preferable interconnection alternative under current technology trend is identified.

The rest of the paper is outlined as follows: Section 2 presents the hierarchical agent monitoring approach and analyzes the features of various monitoring services. Section 3 examines the interconnection requirements to accommodate monitoring communication of different features on the hierarchical agent monitored system, and discusses four interconnection architectures in a theoretic manner. Section 4 presents quantitative evaluation of these alternatives focusing on communication latency, energy and area overhead. Section 5 concludes the paper and discusses the on-going work.

2 Hierarchical Agent Monitored System-on-Chip

Monitoring services are needed to provide system tracing and adjustment at each architectural level. For parallel on-chip systems, hierarchical monitoring agents are proposed to perform all types of monitoring operations. Here we will first examine typical monitoring services in SoCs (Section 2.1), and then present the novel hierarchical agent framework (Section 2.2) .

2.1 Monitoring Services for On-Chip Systems

Many types of monitoring operations, from system-level to circuit-level, have been proposed. They are required for resource management, fault-tolerance, and adjustment of specific run-time parameters.

Application mapping and network configuration are system-level management, which can be performed at run-time given changing application scenarios and system performance. Properly mapping instruction flows onto resources is an effective method to improve performance. For instance, [10] presents efficient algorithms to map processes onto NoC nodes with minimal expected energy consumption under performance constraints. This type of monitoring service is a coarse-grained operation, and performed infrequently while incurring significant amount of reconfiguration overhead. There also exist methods of fine-grained application mapping, which configure a small number of resources to realize a relatively simple processing task. For example, 9 processors can be utilized to realize an JPEG encoder on the 36-processor ASAP chip [11].

Another generic category of monitoring services is fault tolerance, which can be coarse or fine-grained. In a dual or multi-core platform, the failure of a processing core is a coarse-grained error. With the parallelization of on-chip resources, the fault management for a single core or even a communication channel becomes a fine-grained monitoring operation. For instance, [12] describes using spare wires with local reconfiguration circuitry to deal with permanent errors in individual communication channels, and the fine-grained monitoring method incurs small area and power overhead.

Moreover, run-time optimization of specific parameters is another category of monitoring services widely adopted, for instance thermal management and power optimization. Thermal management is usually a coarse-grained operation but based on different thermal conditions specific requirements may be incurred. For instance, the prospect of a thermal breakdown requires an urgent monitoring operation, while normal thermal optimization is a comparatively slow process. In [13], the thermal reconfiguration interval is set as $167\mu s$ (hundreds of thousands of cycles for operating frequency in GHz domain). Power monitoring is more common for embedded systems. Dynamic voltage and frequency scaling (DVFS), as an effective measure to reduce run-time power consumption, has been applied in different granularities. Global-wise or per-cluster DVFS is a conventional technique, but considered inefficient for certain spatially varying traffic patterns [14]. Fine-grained per-core DVFS has been proposed with fast voltage switching and low overhead, for instance [14] uses voltage transition period around $100ns$. Power-gating, which turns off certain components to reduce the static power consumption, is an important technique for parallel systems. Power-gating can be applied in a coarse-grained manner if a large portion of resources are not active for a period of time. It can also be applied to fine-grained circuits with less timing penalty. For instance, [15] presents on/off channels for interconnection networks, and the transition delay as low as 10-100 cycles was considered in the experiment.

Monitoring services, as a stand-alone concept, have been preliminarily studied in few previous works. The work in [16] focuses on the monitoring services for coarse-grained configuration and debugging operations. The configuration flow exemplified contains the messages for setting connections, which is an important yet very small part of the potential monitoring traffics on a general-purpose platform. Two other pioneering works [17, 18] discuss run-time monitoring services targeting NoC-based platform. These works focus on functional analysis of the monitoring traffic and the modification of the design flow with area overhead examined. However, new considerations, for instance power efficiency of monitoring operations and reconfigurability, have rarely been addressed in existing works.

2.2 Hierarchical Agent Monitoring Services

A scalable approach, hierarchical agent monitoring (Fig. 1), is proposed to design run-time monitoring services for parallel on-chip systems. An agent is an embedded module, which traces necessary information and correspondingly configures the assigned components dynamically (Fig.2). The required infor-

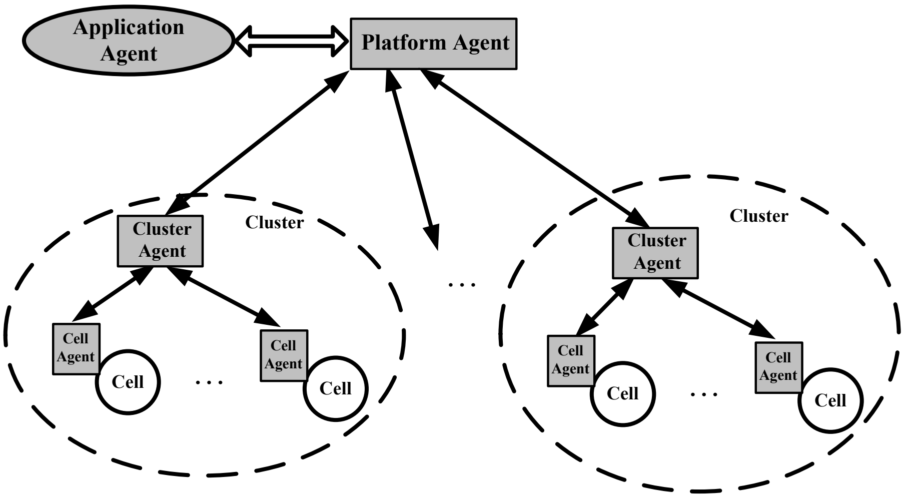


Figure 1. Generic Agent Hierarchy on Parallel Systems

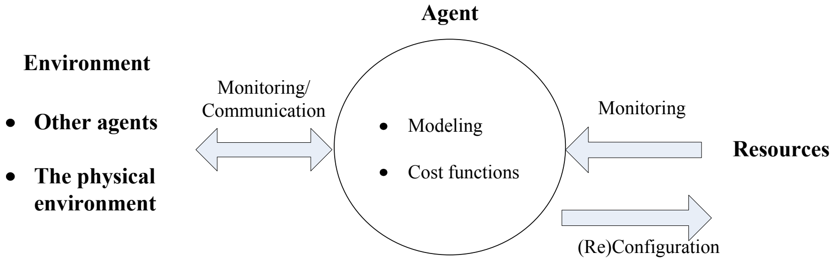


Figure 2. Functional Illustration of Agents

mation may include the status of the assigned components, the messages sent by other agents, and the environment status if relevant to the performance of the monitored components. The monitoring hierarchy is designed based on a hierarchical view of parallel on-chip systems. Such a system commonly consists of a pool of components connected by communication interconnections. A fine-grained component, either a processing unit or a communication component, is identified as a cell. A group of cells can be dynamically configured into a cluster, a relatively coarse-grained unit of resources. A cluster may be assigned for a specific subtask in the application. The whole platform consists of a number of dynamically assigned and configured clusters, as well as the remaining cells which are either spares or broken components.

Hierarchical agents provide monitoring services of various functionalities (Table 1) on each structural level (Fig. 3).

Application agent specifies the application requirements, functional or non-functional, for instance the expected speedup and the affordable peak power, to the platform agent. The platform agent configures coarse-grained global-wise parameters, for instance the network topology and universal switching and routing techniques. It observes the platform performance, for instance the total power consumption of the platform and the average latency of all traffic

Table 1
Major Monitoring Services on Hierarchical Agent Monitored SoCs

Category	Details	Granularity	Monitoring Level	Generic Timing Feature
Configuration	application mapping & network configuration	coarse	platform agent, cluster agent	slow
		fine	cluster agent, cell agent	normal
Fault tolerance	fault management of resources	coarse	platform agent, cluster agent	urgent/ fast
		fine	cluster agent, cell agent	fast
	fault management of agents	coarse	platform agent, cluster agent	urgent/ fast
		fine	cluster agent, cell agent	fast
Thermal Management	thermal breakdown avoidance	coarse	all agents	urgent
	normal thermal optimization	coarse	all agents	normal
Power Optimization	DVFS, Power gating, etc..	coarse	platform agent, cluster agent	slow
		fine	cell agent	normal/ fast

flows. The cluster agent is assigned with finer-grained monitoring operations within a cluster. For instance, the voltage and frequency of a certain cluster can be set specifically in case the program running on the cluster requires a faster/slower processing speed. The cluster agent observes the performance within a cluster, for instance the thermal hotspot and the traffic congestion. The lowest level cell agent provides fine-grained local monitoring. Common operations include error detection and power gating.

Besides performance optimization, hierarchical fault-tolerance is also provided.

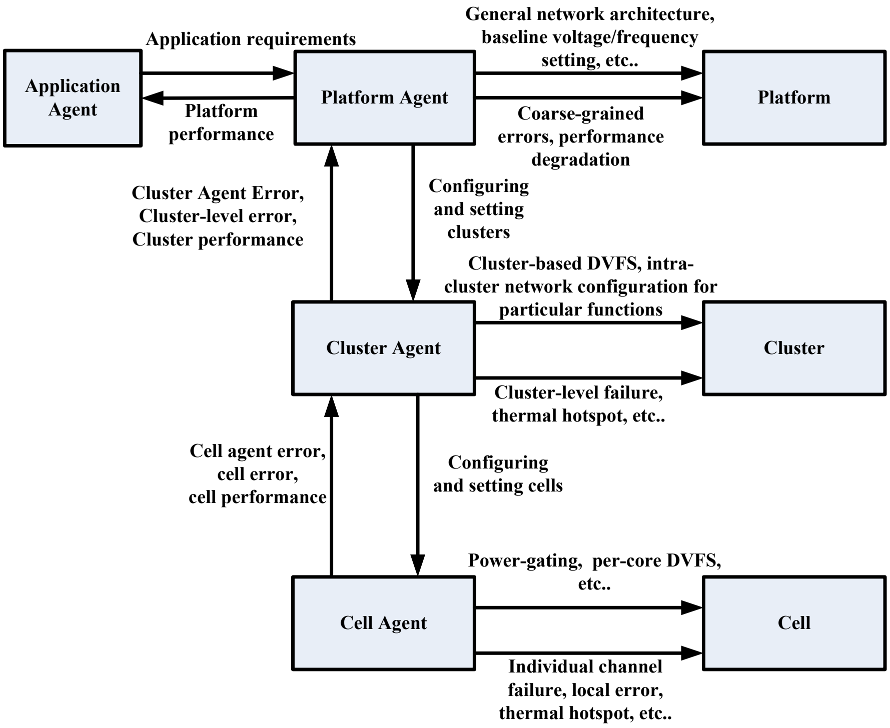

Application Agent
Application requirements
Platform performance
Platform Agent
General network architecture, baseline voltage/frequency setting, etc..
Coarse-grained errors, performance degradation
Platform
Cluster Agent Error, Cluster-level error, Cluster performance
Configuring and setting clusters
Cluster-based DVFS, intra-cluster network configuration for particular functions
Cluster Agent
Cluster
Cluster-level failure, thermal hotspot, etc..
Cell agent error, cell error, cell performance
Configuring and setting cells
Power-gating, per-core DVFS, etc..
Cell Agent
Cell
Individual channel failure, local error, thermal hotspot, etc..

Figure 3. Hierarchical Agent Monitoring Services

Each level of agent monitors the failure of local components and attempts to fix the error itself. If the errors cannot be fixed by the local agent, the errors will be reported to the higher level agent. For instance, transient transmission errors can be fixed by retransmission handled by a monitor on the communication channel [12]. However if whole cells suffer from permanent errors, the high-level topology and routing algorithm may need to be reconfigured [19], which belongs to the responsibility of the platform agent.

The agents themselves are also victims of errors. The failure of agents makes the corresponding components non-observable. The failure of the platform agent or a cluster agent significantly influence the system operation considering the amount of resources it is in charge of. This failure needs to be immediately detected and handled. The cell agent failure, though being less critical, still requires fast handling as the local data processing and communication are seriously influenced with unpredictable state. To deal with agent errors, higher level agent regularly checks the status of lower level agent. When one agent fails, there can be various mechanisms to ensure proper running of the application. One alternative is to incorporate backup agents, especially on the level of application and platform agent, where catastrophic consequences may happen in case of agent failure. Another alternative is to dynamically assign resources to a healthy agent. The detailed discussion of fault-tolerance in hierarchical agent monitored platform is beyond the scope of this paper.

3 Interconnection Architectures for Hierarchical Monitoring Communication

Interconnection architectures require careful analysis to support hierarchical monitoring communication. The study is driven by the general concept of quality-of-service in on-chip communication (Section 3.1), where monitoring communication is treated as a prioritized traffic class over data traffic. We will examine the requirements for this type of traffic class (Section 3.2), and present several potential interconnection architectures (Section 3.3).

3.1 Quality of Service for On-chip Communication

Quality of service (QoS), in terms of network communication, refers to the methods of providing the required performance to specific network clients, for instance a type of traffic [20]. There are generically two types of QoS services, best-effort and guaranteed communication. Best-effort service, which provides no special treatment to individual traffic classes, has the lowest design complexity, while being not able to offer predictable and guaranteed performance. Guaranteed services, on the other hand, ensure certain metrics of performance for a specific traffic class, for instance guaranteed bandwidth or bounded latency. Usually the provision of guaranteed services requires the constraints on the traffic itself, for example an upper boundary on the maximal traffic load.

There exist various methods to provide guaranteed QoS in on-chip communication. TDMA, CDMA (code division multiple access) and physically separate networks are three widely-used techniques. TDMA reserves time slots for a specific traffic class, usually combined with buffer space reservation [21]. It requires modification of the switching fabrics [22] to offer virtual channel arbitration. CDMA method decouples traffic classes by using orthogonal spreading codes. Code generators and demodulators [23] are needed in transmitters and receivers respectively. Physically separate networks, instead of using virtual channels, decouple traffic classes by using dedicated links. Silicon area is sacrificed for simpler switching and arbitration. For instance the TILE64 processor incorporates 5 physically separate networks for different types of traffics [24].

Monitoring communication is a special traffic class to enable monitoring operations. Monitoring services, as used to trace and adjust system status in case of failure and performance modification (Table 1), are performed constantly during application execution. Thus, the monitoring communication needs to be treated with guaranteed services, decoupled from the data traffic. Several previous works have addressed the issues of QoS for certain monitoring communication flows. The work in [7] identifies signaling traffic along with

Table 2
Priorities of Monitoring Communication of Different Timing Features

Service Type (Timing feature)	1st Priority	2nd Priority	3rd Priority
Urgent	extremely fast connection	energy efficiency	area overhead
fast/normal	predictable and guaranteed latency	energy efficiency	area overhead
slow	energy efficiency	predictable latency	area overhead

3 types of data traffics. The signaling traffic it considers covers urgent messages which should have the highest priority and very fast connection. Routing and switching microarchitectures to provide guaranteed transmission targeting asynchronous systems are studied in [21], though some analysis is applicable to synchronous systems as well. In the context of hierarchical monitoring services, various types of communication may be transmitted on different level of interconnection. Thus the differentiating these types is necessary and also beneficial since tailored configurations can be applied.

3.2 Requirements for Hierarchical Monitoring Communication

A general-purpose platform is expected to experience various types of monitoring services. Each of these services have requirements with different priorities based on their performance features. Table 2 specifies the priorities of each service type characterized by its timing features (details of these services can be found in Table 1). Based on these priorities, several requirements on the interconnection architectures can be identified.

- Predictable and Guaranteed Latency

Urgent messages requires guaranteed low latency, and other monitoring communication flows also need predictable latencies. Since the data traffic is difficult to be bounded in all temporal periods for any potential application, the latency of monitoring communication needs to be decoupled from that of data traffic. Reservation of bandwidth is an effective method to achieve predictable and guaranteed average latency. Strictly speaking, additional arbitration is needed to theoretically guarantee the maximum latency of an individual message. Though when assuming a fair switching arbitration, before the network saturates, the likelihood of a significantly delayed message is small. For example, [7] studies the delay of 99%-99.9% of the packets. For very urgent and critical messages, extra arbitration may still be needed. This issue will be studied in our future work, and here we use guaranteed bandwidth to provide predictable and bounded average latency with the assumption that the monitoring communication never saturates the maximum bandwidth allocated to it.

- High Energy Efficiency

Despite the relatively low volume of monitoring traffic compared to common data traffics, the energy efficiency is still a highly prioritized requirement. With more fine-grained monitoring operations in massively parallel on-chip systems, the traffic volume is expected to increase significantly. In addition, the global interconnect consumes considerable amount of power, which will be incurred by monitoring operations across multiple agent levels. Moreover, the encoding and transmission manners of monitoring messages influence the amount of payload of the monitoring traffic [16]. For certain ultra-low-power applications, for instance sensor networks, the monitoring flow will be the major source of power consumption when the data traffic becomes very low, as the monitoring services should not be turned off otherwise the status of the system will be non-observable.

- Reconfigurability

Considering the different features of various monitoring communication flows, reconfiguration is needed not only to fulfill the transmission requirements, but also to achieve lower overhead. For example, for slow monitoring operations, the interconnection can be configured with low operating frequency to reduce the power consumption as long as the timing requirement is still met. When urgent operations appear, the channel should be configured with high speed.

- Affordable Area Overhead

This conventional design constraint is alleviated in current and emerging on-chip systems, since quite abundant wirings can be provided by the state-of-the-art multi-layer fabrication process [24]. Nonetheless, the area overhead should still be made small and affordable.

3.3 Interconnection Alternatives for Hierarchical Monitoring Communication

Based on the previously outlined requirements, several generic architectures can be introduced to support the monitoring communication:

- Baseline Best-Effort Interconnection

When the agent communication is overlapped with data communication without special treatment, this alternative is considered as a best-effort interconnection architecture, used as the baseline for comparison with guaranteed services.

The baseline architecture suffers from several performance weaknesses. Monitoring communication is coupled with data traffic, which leads to unpredictable latency. When the network faces heavy traffic load, the latency increases significantly and the agent communication will suffer from similar latency increase as the data traffic. In addition, the channels can not be flexibly configured as the data communication will be influenced as well. The benefit of the best-effort interconnection alternative is low switching complexity and reduced area overhead for wiring.

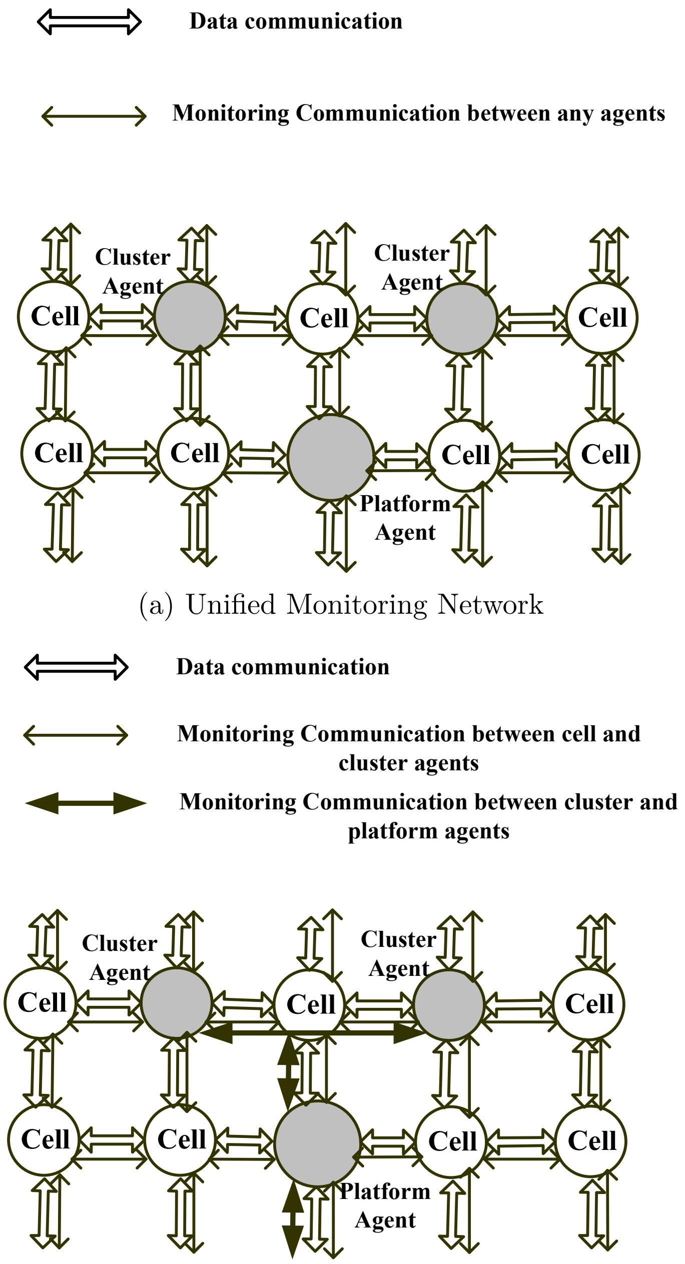
- TDMA-based Interconnection

A conventional manner of guaranteed service is TDMA-based communication. Timeslots can be reserved for the monitoring communication. The assigned timeslots can only be utilized for data communication if there is no monitoring communication present in the slot of the switch. Buffers are allocated to data and monitoring communication separately, in order to decouple the two traffic classes into virtual channels. TDMA-based connection provides guaranteed bandwidth with moderate reconfigurability. The silicon area is increased compared to the baseline alternative since the switches are more complicated. More importantly, virtual channeling increases the energy consumption significantly for every traversal in the switching fabric.

- Physically Separate Networks

Physically decoupling the agent monitoring communication from data traffic results in physically separate networks. Specific to the hierarchical agent monitoring framework, two alternatives of physical decoupling are applicable (Fig. 4(a) and (b)) The first alternative is unified monitoring network (Fig. 4(a)), where all monitoring communications are transmitted on a physically separate network decoupled from the data communication network. The other alternative is hierarchical monitoring networks (Fig. 4(b)). In this architecture, two monitoring networks are decoupled from the data traffic. One of them is used for communication between cluster and cell agents, and the other one is dedicated to the communication between cluster agents and the platform agent.

Physically separate networks significantly reduces the switching and arbitration complexity in the switching fabric, which provides high energy efficiency while costing more wiring overhead. It allows the maximal flexibility in configuring the networks adaptively based on the monitoring traffics on different architectural levels. Hierarchical monitoring networks, as a design



(a) Unified Monitoring Network

(b) Hierarchical Monitoring Networks

Figure 4. Physically Separate Monitoring Networks

choice based on the hierarchical agent monitoring framework, further decouple coarse-grained and fine-grained monitoring communication onto two networks. This method considers the typical differences in the monitoring communication between the platform/cluster agents, and that between the cluster/cell agents. As summarized in Table 1, higher-level monitoring communication between the platform and cluster agents usually concerns coarse-grained configuration with long timing interval, while lower-level communication between the cluster and cell agents concerns fine-grained configuration with urgent timing requirements. The separation between these two levels of communica-

Table 3
Qualitative Comparison of Interconnection Alternatives for Hierarchical Monitoring Communication

	Type of QoS	Flexibility of Configuration	Latency	Energy Consumption	Area Overhead
baseline	best-effort	low	dependent on data traffic	high	lowest
TDMA	guaranteed-service (reserved bandwidth & buffer space)	medium	bounded average latency	highest	low
unified monitoring network	guaranteed-service (physically independent switching)	high	bounded average latency	low	high
hierarchical monitoring networks	guaranteed-service (physically independent switching)	highest	bounded average latency	lowest	highest

tion enables flexible settings for each of them, so that power efficiency can be maximized while their own timing requirements can be met.

The qualitative features of each presented interconnection alternative are summarized in Table 3, and quantitative evaluation will be presented in Section 4.

4 Quantitative Evaluation on NoC-based Platform

Here we quantitatively evaluate the four interconnection architectures for monitoring communication (Section 3.3) on the NoC platform, which is the most promising structure for parallel SoC systems. Based on 65nm power model and NoC microbenchmarks, the communication latency, energy efficiency and area overhead are examined, leading to the design choice of the most appropriate interconnection alternative for hierarchical monitoring communication.

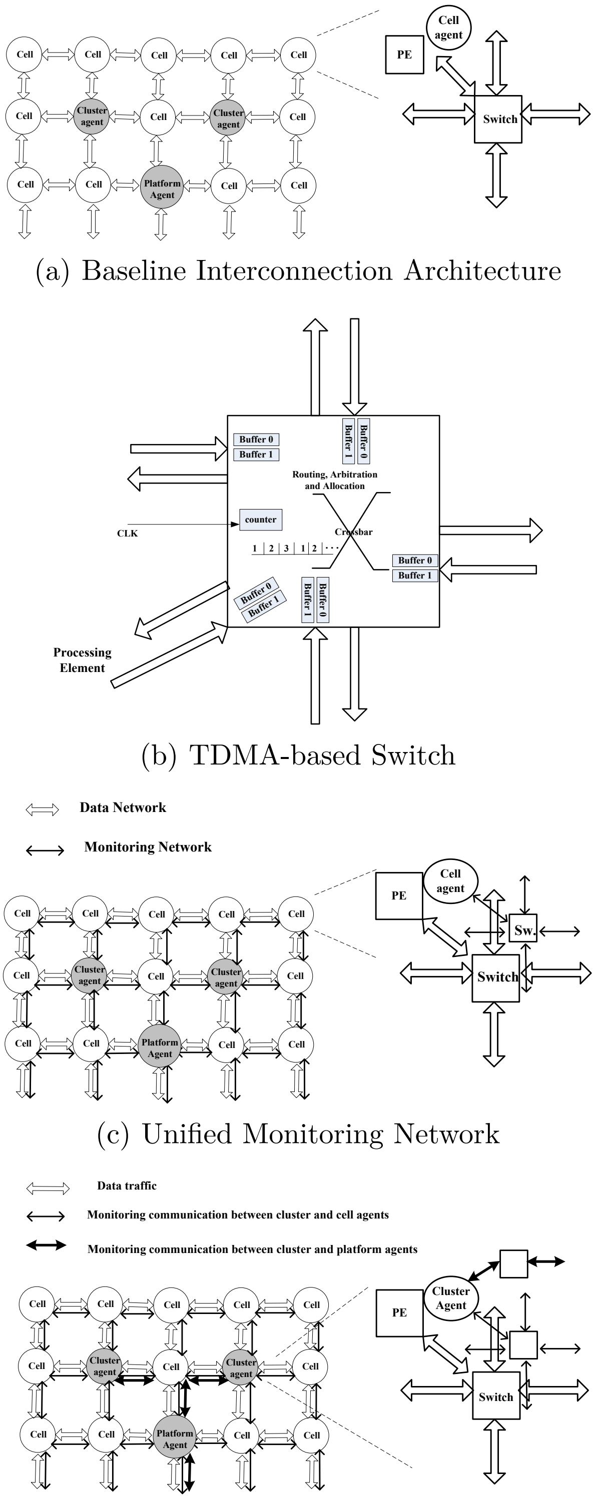
The baseline architecture is an $8 * 8$ mesh-based NoC (Fig 5(a)). On the modularized structure, hierarchical agents can be conveniently mapped. A processing element with its switch, including necessary interfaces, as well as the channels starting from the switch, can be identified as a cell. The cell agent is to be located within the geographic area of the cell, so that the monitoring connection within a cell is local wiring with negligible latency and energy overhead. A number of cells in the network are dynamically configured into a cluster, and the cluster agent is hosted by a processing element. The platform agent is performed by a dedicated processor in the network. Best-effort interconnection uses the existing data channel for monitoring communication without any prioritized arbitration (Fig. 5(a)).

TDMA-based interconnection architecture uses the same physical network as the baseline architecture, but one out of three timeslots gives priority to the monitoring communication. Only when there is no monitoring traffic will the timeslot be used for data switching. The switches use double sets of buffers with one set reserved for the monitoring communication (Fig. 5(b)).

The interconnection alternative with unified monitoring network uses two separate physical networks (Fig. 5(c)), with one network dedicated to the monitoring communication between any level of agents. Thus each node incorporates two sets of switches.

The alternative with hierarchical monitoring networks incorporates two monitoring networks (Fig. 5(d)). One of them is for monitoring communication between cell and cluster agents, which is a mesh-based network connecting the cell and cluster agents on all node. The other is a specialized network only connecting the cluster and platform agents, dedicated to the high-level monitoring communication.

The networks use store-and-forward single-flit switching. In the specialized monitoring network, the routing path is unique because of the limited connectivity. In the fully-connected mesh networks, X-Y routing is used. The locations of cluster agents and the platform agent for experimental purposes are illustrated in Fig. 6 considering reasonable geographic symmetry. It should be noted that the generic comparison between the interconnection architectures is independent from the exact flow control algorithm and locations of agents.



(a) Baseline Interconnection Architecture

(b) TDMA-based Switch

(c) Unified Monitoring Network

(d) Hierarchical Monitoring Networks

Figure 5. Interconnection Architectures for Hierarchical Monitoring Communication on NoCs

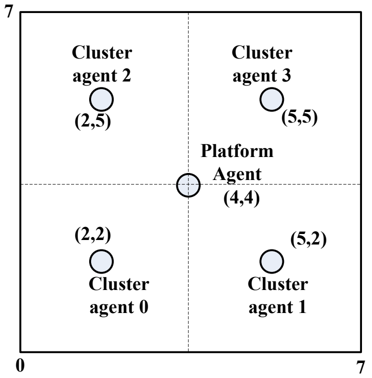


Figure 6. Positions of Cluster and Platform Agents on the Experimental NoC Platform

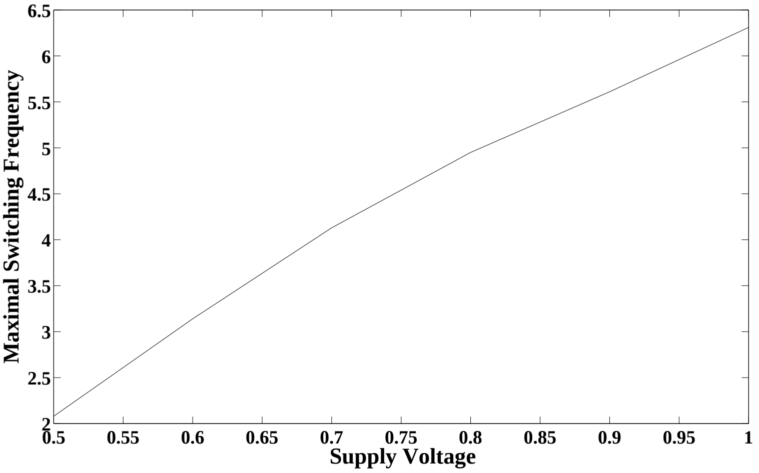


Figure 7. Voltage/Frequency Relation for Inter-Switch Links ($1mm$, distributed RLC model, extracted from simulations in Cadence Spectre)

4.2 Power Estimation

Power estimation of on-chip communication is enabled by Cadence simulation and external Orion 2.0 tool [25]. We modeled the inter-switch links in Cadence Spectre using 65nm STMicroelectronics technology, and simulated the maximal switching frequencies within a range of supply voltages (Fig. 7). To enable tradeoff analysis of different network settings, one high voltage/frequency pair (6GHz, 1.0V) and a low voltage/frequency pair (2GHz, 0.5V) are extracted, with the voltage values slightly higher than the minimal figures in the curve to give proper design slack. It should be noted that the pairs are chosen for experimental purposes, and in practice, the voltage and frequency should be configured based on the specific platform and application requirements.

Given the two pairs of voltage and frequency settings, the energy consumption of the interconnection components can be estimated by Orion 2.0. This

Table 4
Experimental Traffic Traces

Index	Traffic Pattern	Trace Detail
T0	uniform	uniform destination, low traffic
T1	uniform	uniform destination, high traffic
T2	locality	locality destination, the region in (0,3,0,2) has high traffic
T3	hotspot	70% of packets are destined to the region (0,3,0,2)

tool provides accurate system-level calculation of power consumption for on-chip switches and links, by using detailed modeling equations and technology parameters [25]. With supply voltage and working frequency specified, the energy consumed by flits traversing the interconnection can be estimated. Each switch is modeled with the input-buffered structure (Fig. 5(b)), with matrix crossbar and register-type buffers. Each link is $1mm$ long, as intermediate wirings. For TDMA-based connection, two virtual channels are integrated. The monitoring networks are assumed as 8-bit wide. Other parameters are set as default values in Orion tool, using 65nm technology.

4.3 Synthetic Traffic Patterns

NoC microbenchmarks characterize traffic patterns potentially experienced in on-chip networks, and are suitable for early-stage architectural explorations of interconnection design [26, 27]. To evaluate the presented interconnection alternatives, four traffic traces generated from NoC microbenchmarks [27] are used in the experiments, categorized into three types of traffic patterns: uniform, locality and hotspot (Table 4).

Uniform traffic pattern assumes that every node has the same probability to be the transmission destination. T0 and T1 are both uniform traffics though with different amount of traffics, in order to evaluate the influence of temporal traffic variation.

Locality pattern models the network traffic where adjacent nodes have higher probability of mutual transmission. The pattern is modeled by Eq. 1, where $P(d)$ is the probability of transmission destined to a node with the distance of d. $A(D)$ is the normalizing factor that makes sure the probabilities sum up to 1. D is the maximum distance in the network. Existing network mapping algorithms, for instance [10], locate heavily communicating processes onto nearby processing units, so that the total energy consumption can be minimized because of shorter communication distances. With such mapping algorithms, locality traffic traces are likely to appear in the network.

$$\begin{aligned} P(d) &= 1/(A(D) * 2^d) \\ A(D) &= \sum_{d=1}^{D} (1/2^d) \end{aligned} \tag{1}$$

Hotspot pattern assigns a high transmission probability to certain regions of the network. Such pattern of communication is likely to appear when certain processors are major data consumers in a parallel computing platform. Eq. 2 and Eq. 3 model the probabilities of a node in the hotspot region and other region as the transmission destination respectively. ρ is the fraction of the traffic targeted to the hotspot region. $N_{hotspot}$ is the number of nodes in the hotspot region. $N_{network}$ is the total number of nodes in the network.

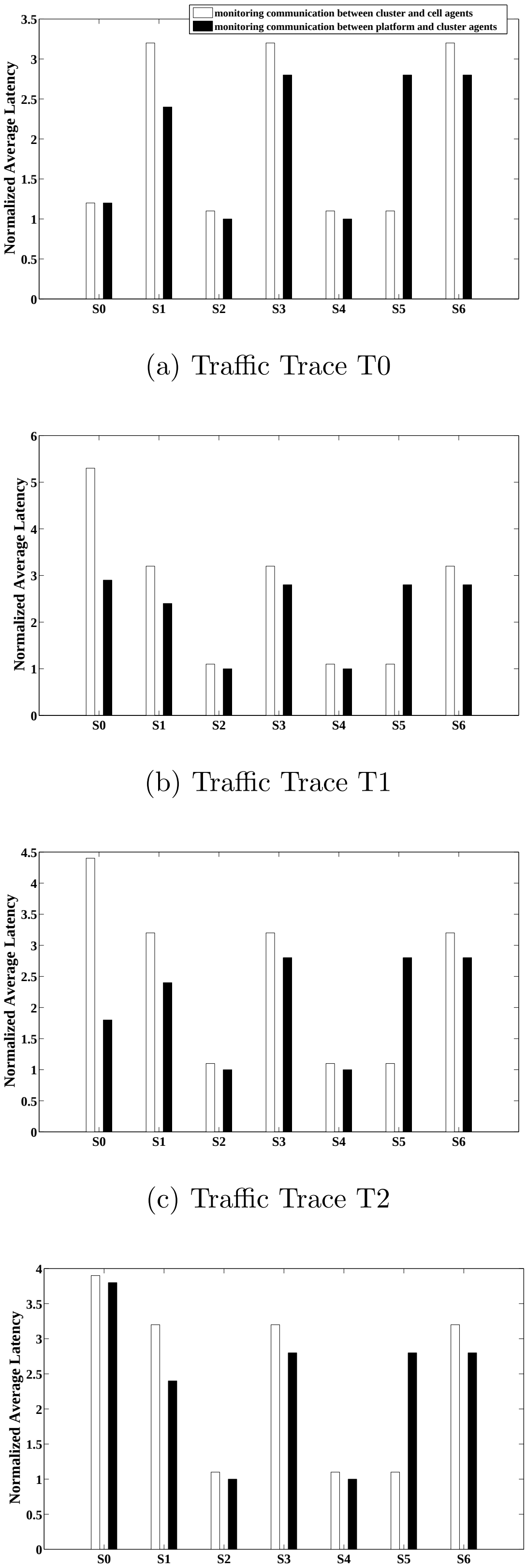
$$P(hotspot) = \rho / N_{hotspot} \tag{2}$$

$$P(other) = (1 - \rho)/(N_{network} - N_{hotspot}) \tag{3}$$

4.4 Simulation Results

To comprehensively analyze the features of the four interconnection alternatives, seven settings (Table 5) are simulated running each of the four network traces (Table 4). In Table 5, “HP” stands for the high voltage/frequency pair, and “LP” stands for the low voltage/frequency pair (Section 4.2). The 1st monitoring network refers to the mesh network for monitoring communication between the cell and cluster agents, and the 2nd monitoring network refers to the specialized network for monitoring communication between the cluster and platform agents (Section 4.1).

Average transmission per-flit latency of each trace (Table 4) under the seven network settings (Table 5) is summarized in Fig. 8. The latency of monitoring communication between cell and cluster agents is measured in the area with highest data traffic, for traces with spatial variation (T2 and T3). The values are normalized with the lowest latency measured in the experiments, so that they can be compared across different traffic traces. We can observe that the monitoring communication latency is heavily influenced by the data traffic in the baseline architecture (S0), as the latencies are large in heavy data traffic (for instance T1 and T3) while reasonably small in light data traffic (T0). In TDMA-based interconnection (S1), the latency is relatively high but not dependent on the data traffic pattern. The latencies for the two types



(a) Traffic Trace T0

(b) Traffic Trace T1

(c) Traffic Trace T2

(d) Traffic Trace T3

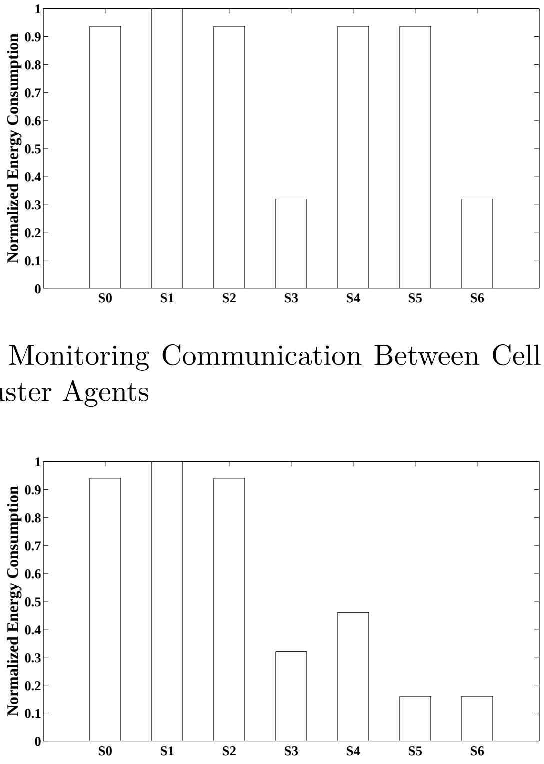
Figure 8. Normalized Transmission Latency of Monitoring Communication in Different Network Settings

Table 5
Experimental Network Settings

Index	Architecture	Power Supply
S0	Baseline	HP
S1	TDMA	HP
S2	Unified Monitoring Network	HP for data network HP for monitoring network
S3	Unified Monitoring Network	HP for data network LP for monitoring network
S4	Hierarchical Monitoring Networks	HP for data network HP for the 1st monitoring network HP for the 2nd monitoring network
S5	Hierarchical Monitoring Networks	HP for data network HP for the 1st monitoring network LP for the 2nd monitoring network
S6	Hierarchical Monitoring Networks	HP for data network LP for the 1st monitoring network LP for the 2nd monitoring network

of monitoring communication remain constant in each traffic trace since the bandwidth is ensured. Unified and hierarchical monitoring networks (S2-S6) provide low-latency transmission for the monitoring communication, as being decoupled from the data traffic. In particular, in hierarchical monitoring networks, the transmission latency can be set specifically to the requirements of each level of monitoring communication. For example, if the monitoring communication between the cell and cluster agents requires fast connection, while the higher level communication between the cluster and platform agents allows for longer delay, setting S5 can be configured.

Average per-flit energy consumption of each network setting is summarized in Fig. 9. The measurement of energy consumption is not influenced by the traffic patterns since the number of transmission hops is the same (as we use minimal routing). The values are normalized with the that of the most energy consuming setting for the specific level of monitoring communication. We can observe from Fig. 9 that TDMA-based interconnection consumes the highest energy, while unified and hierarchical monitoring networks provide significant benefits in energy efficiency. Such improvement originates from the difference in energy consumption of flits traversing each type of routers (Fig. 10). Hierarchical monitoring networks, in particular, considerably reduce the energy consumption of the communication between cluster and platform agents, since



(a) Monitoring Communication Between Cell and Cluster Agents

(b) Monitoring Communication Between Cluster and Platform Agents

Figure 9. Normalized Energy Consumption of Monitoring Communication in Different Network Settings

the specialized network uses low-degree crossbars (Fig. 5 (d)), which consume the lowest energy (Fig. 10). In addition, hierarchical monitoring networks allow configurable tradeoff on the two levels of monitoring communication. For example, if the coarse-grained monitoring service on clusters issued by the platform agent is latency-tolerant, the monitoring communication between platform and cluster agents can be set with low power configuration while leaving the low level monitoring communication with fast connection (S5).

The area estimation of each network architecture is summarized in Table 6. The area of each network component is obtained from Orion 2.0. The data network uses 32-bit wide (per-direction) links, and monitoring channels in physically separate networks use 8-bit wide (per-direction) links, as the monitoring communication is typically lower in volume than the major data flow.

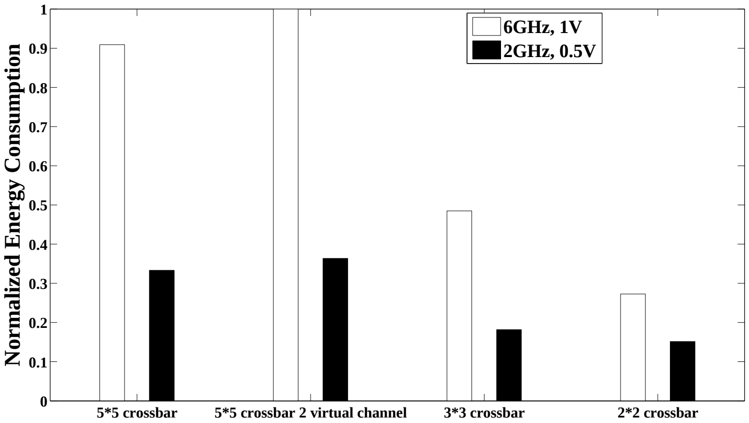


Figure 10. Normalized Energy Consumption of Flits Traversing Different Routers (obtained from Orion 2.0)

Table 6
Area Estimation for Each Interconnection Architecture ($8 * 8$ mesh)

Architecture	Switches (mm^2)	Wires (mm^2)	Total (mm^2)	% of a chip area $275mm^2$ (TeraFLOPS)
Baseline	7.2	4.5	11.7	4.3%
TDM	7.4	4.5	11.9	4.3%
Unified Monitoring Network	7.7	5.6	13.3	4.8%
Hierarchical Monitoring Networks	7.7	5.7	13.4	4.9%

4.5 Quantitative Experiment Analysis

The experiments validate the benefits and issues of each interconnection alternative as theoretically reasoned in Section 3.3. In baseline architecture with best-effort service, the monitoring communication suffers from large latency in heavy data traffic caused by temporal and spatial variations. The benefit of the baseline architecture is smaller silicon area. TDMA-based interconnection incurs the most energy consumption, while keeping a constant latency in any type of traffic pattern with reserved bandwidth for the monitoring communication. Physically separate networks, in general, have higher energy efficiency with simpler switching process. The latency of monitoring communication is independent of the data traffic, and adjustable by configuring different network frequencies.

In particular, hierarchical monitoring networks are demonstrated as the most suitable architecture on the hierarchical agent monitored platform. In terms of transmission latency, this alternative can achieve as high as 79% latency reduction for the monitoring communication between cell and cluster agents (in trace T1) and 74% reduction for the higher level monitoring communication (in trace T3), when configured in high power setting (S4), compared to the baseline architecture. In terms of energy efficiency, hierarchical monitoring networks reduce the average energy consumption significantly compared to the TDMA-based network. When configured in low power setting (S6), the energy reduction is 68% and 84% for the two levels of monitoring communication respectively. The energy reduction is more profound for the high level monitoring communication since the specialized network with dedicated connections between platform and cluster agents integrate low-degree switches. Moreover, hierarchical monitoring networks allow flexible configuration on each level of monitoring communication (S4-S6). Given different requirements of monitoring flows of various granularities (Table 1), such flexible settings are needed to support multiple monitoring services on the platform. The area overhead of separate monitoring networks (Table 6) is moderate considering the constant technology progress in multi-layer chip fabrication.

5 Conclusion

This article presented a system-level analysis of the interconnection architectures suitable for a novel design platform. Hierarchical agent monitored System-on-Chip provides a scalable solution to the design of massively parallel on-chip systems with variability and adaptivity. Proper interconnection architectures are needed to support the monitoring communication, which is prioritized over data traffics and requires guaranteed transmission.

We first presented the hierarchical agent monitoring approach, focusing on various monitoring operations with different functions, granularities and timing features. These monitoring operations are assigned to specific levels of agents, which result in hierarchical monitoring communication in the system. We analyzed the priorities of the communication flows, and outlined the requirements of suitable interconnection to support the monitoring communication. Based on these requirements, we examined several generic architectures, both theoretically and quantitatively, in terms of the transmission latency, energy efficiency and area overhead. The quantitative experiments were built on an 8*8 mesh-based network-on-chip platform, with power estimation enabled by Cadence simulation and Orion 2.0 tool.

From the study, we found that physically separate networks provide flexible and energy-efficient transmission for monitoring communication, with guaranteed latency independent of data traffic conditions. In particular, hierarchical monitoring networks are the most appropriate solution on the platform with hierarchical agents. As this article addresses generic discussion of interconnection architectures, we are currently working on specific applications and systems. After mapping these applications onto hierarchical agent monitored platforms, we can examine the presented architectures with case-dependent results.

Acknowledgements

The authors have been supported by the Foundation of Nokia Corporation.

References

[1] D.N. Truong, W.H. Cheng, T. Mohsenin, Zhiyi Yu, A.T. Jacobson, G. Landge, M.J. Meeuwsen, C. Watnik, A.T. Tran, Zhibin Xiao, E.W. Work, J.W. Webb, P.V. Mejia, and B.M. Baas. A 167-processor computational platform in 65 nm cmos. *IEEE Journal of Solid-state Circuits*, 44(4):1130–1144, 2009.

[2] Shekhar Borkar. Designing reliable systems from unreliable components: The challenges of transistor variability and degradation. *IEEE Micro*, 25(6):10–16, 2005.

[3] Pekka Rantala, Jouni Isoaho, and Hannu Tenhunen. Novel agent-based management for fault-tolerance in network-on-chip. In *Proc. 10th Euromicro Conference on Digital System Design Architectures, Methods and Tools DSD 2007*, pages 551–555, 2007.

[4] Alexander Wei Yin, Liang Guang, Pasi Liljeberg, Pekka Rantala, Ethiopia Nigussie, Jouni Isoaho, and Hannu Tenhunen. Hierarchical agent architecture for scalable noc design with online monitoring services. 1st International Workshop on Network on Chip Architectures (in conjunction with MICRO41), 2008.

[5] K. Keutzer, A.R. Newton, J.M. Rabaey, and A. Sangiovanni-Vincentelli. System-level design: orthogonalization of concerns and platform-based design. *IEEE Transactions on CAD*, 19(12):1523–1543, Dec. 2000.

[6] Cristina Aurrecoechea, Andrew T. Campbell, and Linda Hauw. A survey of qos architectures. *Multimedia Systems Journal, Special Issue on QoS Architecture*, 6:138–151, 1996.

[7] Evgeny Bolotin, Israel Cidon, Ran Ginosar, and Avinoam Kolodny. Qnoc:

Qos architecture and design process for network on chip. *Journal of Systems Architecture*, 50(2-3):105–128, 2004.

[8] Krste Asanovic, Ras Bodik, Bryan Christopher Catanzaro, Joseph James Gebis, Parry Husbands, Kurt Keutzer, David A. Patterson, William Lester Plishker, John Shalf, Samuel Webb Williams, and Katherine A. Yelick. The landscape of parallel computing research: A view from berkeley. Technical report, U.C.Berkeley, 2006.

[9] Jan M. Rabaey. Scaling the power wall: Revisiting the low-power design rules. Keynote speech at SoC 07 Symposium, Tampere, November 2007.

[10] Jingcao Hu and R. Marculescu. Energy and performance-aware mapping for regular noc architectures. *IEEE Transactions on CAD*, 24(4):551–562, 2005.

[11] Zhiyi Yu, M. Meeuwsen, R. Apperson, O. Sattari, M. Lai, J. Webb, E. Work, T. Mohsenin, M. Singh, and B. Baas. An asynchronous array of simple processors for dsp applications. In *Proc. Digest of Technical Papers. IEEE International Solid-State Circuits Conference ISSCC 2006*, pages 1696–1705, 2006.

[12] Teijo Lehtonen, Pasi Liljeberg, and Juha Plosila. Online reconfigurable self-timed links for fault tolerant noc. *VLSI Design*, 2007:13, 2007.

[13] P. Chaparro, J. Gonzalez, G. Magklis, Cai Qiong, and A. Gonzalez. Understanding the thermal implications of multi-core architectures. *IEEE Transactions on Parallel and Distributed Systems*, 18(8):1055–1065, 2007.

[14] Hyunjin Kim, Hyejeong Hong, Hong-Sik Kim, Jin-Ho Ahn, and Sungho Kang. Total energy minimization of real-time tasks in an on-chip multiprocessor using dynamic voltage scaling efficiency metric. *IEEE Transactions on CAD*, 27(11):2088–2092, Nov. 2008.

[15] V. Soteriou and Li-Shiuan Peh. Exploring the design space of self-regulating power-aware on/off interconnection networks. *IEEE Transactions on Parallel and Distributed Systems*, 18(3):393–408, 2007.

[16] C. Ciordas, T. Basten, A. Radulescu, K. Goossens, and J. Meerbergen. An event-based network-on-chip monitoring service. In *Proc. of IEEE High-Level Design Validation and Test Workshop*, pages 149–154, 2004.

[17] C. Ciordas, K. Goossens, A. Radulescu, and T. Basten. Noc monitoring: impact on the design flow. In *Proc. IEEE International Symposium on Circuits and Systems ISCAS 2006*, pages 1981–1984, 2006.

[18] C. Ciordas, K. Goossens, T. Basten, A. Radulescu, and A. Boon. Transaction monitoring in networks on chip: The on-chip run-time perspective. In *Proc. International Symposium on Industrial Embedded Systems IES '06*, pages 1–10, 2006.

[19] Lei Zhang, Yinhe Han, Qiang Xu, and Xiaowei Li. Defect tolerance in homogeneous manycore processors using core-level redundancy with unified topology. In *Proc. Design, Automation and Test in Europe DATE '08*, pages 891–896, 2008.

[20] William James Dally and Brian Towles. *Principles and practices of interconnection networks*. Morgan Kaufmann, 2004.

[21] T.Felicijan and S.B.Furber. Quality of service (qos) for asynchronous on-chip networks. In *Formal Methods for Globally Asynchronous Locally Synchronous Architecture (FMGALS 2003)*, 2003.

[22] Kees Goossens, John Dielissen, Jef van Meerbergen, Peter Poplavko, Andrei Rădulescu, Edwin Rijpkema, Erwin Waterlander, and Paul Wielage. *Networks on chip*, chapter Guaranteeing The Quality of Services in Networks on Chip, pages 61–82. Kluwer Academic Publishers, Hingham, MA, USA, 2003.

[23] Daewook Kim, Manho Kim, and G.E. Sobelman. Cdma-based network-on-chip architecture. In *Proc. IEEE Asia-Pacific Conference on Circuits and Systems*, volume 1, pages 137–140 vol.1, 2004.

[24] D. Wentzlaff, P. Griffin, H. Hoffmann, Liewei Bao, B. Edwards, C. Ramey, M. Mattina, Chyi-Chang Miao, J.F. Brown, and A. Agarwal. On-chip interconnection architecture of the tile processor. *IEEE MICRO*, 27(5):15–31, 2007.

[25] A.B. Kahng, Bin Li, Li-Shiuan Peh, and K. Samadi. Orion 2.0: A fast and accurate noc power and area model for early-stage design space exploration. In *Proc. DATE '09*, pages 423–428, 2009.

[26] C. Grecu, A. Ivanov, R. Pande, A. Jantsch, E. Salminen, U. Ogras, and R. Marculescu. Towards open network-on-chip benchmarks. In *Proc. First International Symposium on Networks-on-Chip NOCS 2007*, pages 205–205, 2007.

[27] Z. Lu, A. Jantsch, E. Salminen, and C. Grecu. Network-on-chip benchmarking specification part 2: Microbenchmark specification version 1.0. Technical report, OCP International Partnership Association, Inc., May 2008.