Des Autom Embed Syst (2015) 19:59–76
DOI 10.1007/s10617-014-9137-6

The fast evolving landscape of on-chip communication

Selected future challenges and research avenues

**Davide Bertozzi · Giorgos Dimitrakopoulos ·
José Flich · Sören Sonntag**

Published online: 30 April 2014

Abstract As multi-core systems transition to the many-core realm, the pressure on the interconnection network is substantially elevated. The Network-on-Chip (NoC) is expected to undertake the expanding demands of the ever-increasing numbers of processing elements, while—at the same time—technological and application constraints increase the pressure for increased performance and efficiency with limited resources. Although NoC research has evolved significantly the last decade, essential questions remain un-answered and call for fresh research ideas and innovative solutions. In this paper, we summarize a selected set of NoC-related research challenges, with the hope to guide future development and trigger high-impact research progress.

Keywords Network on chip · Microarchitecture · Physical integration · Optical NoCs · Wireless NoCs · Virtualization

1 Introduction

Modern integrated multicore platforms have adopted a Network-on-Chip (NoC) technology that brings interconnect architectures inside the chip. The NoC paradigm tries to find a

D. Bertozzi (✉)
University of Ferrara, Ferrara, Italy
e-mail: davide.bertozzi@unife.it

G. Dimitrakopoulos
Democritus University of Thrace, Xanthi, Greece
e-mail: dimitrak@ee.duth.gr

J. Flich
Universitat Politècnica de València, Valencia, Spain
e-mail: jflich@disca.upv.es

S. Sonntag
Intel Corp, Munich, Germany
e-mail: soeren.sonntag@intel.com

scalable solution to the tough integration challenge of modern SoCs, by applying at the silicon chip level well established networking principles, after suitably adapting them to the silicon chip characteristics and to application demands [1–4]. This approach was originally adopted to tackle the physical integration complexity, clocking scalability, timing closure and verification problems of state-of-the-art SoCs [5]. While the seminal idea of applying networking technology to address the chip-level interconnect problem has been shown to be adequate for current systems, the complexity of future computing platforms demands new architectures that go beyond physical-related requirements and equally participate in delivering high-performance, quality of service, dynamic adaptivity at the minimum energy and area overhead [6].

Scalable interconnect architectures form the solid base on top of which heterogeneous computing platforms and their unifying programming environments will be developed. Parallelism is all about cooperation that cannot be achieved without the efficient communication offered by the interconnect. The interconnect implements the physical and logical medium for any kind of data transfer and its latency, bandwidth and energy efficiency directly affects overall system performance. Interconnect design is a multidimensional problem involving hardware and software components such as network interfaces, switches, topologies, routing algorithms and communication library APIs. We expect system networks to achieve ultra-fast end-to-end message delivery, hundreds of Gbytes per second of link under demanding physical, architectural and technological constraints which translate to contradictory objectives and most commonly to very tight energy budgets.

In this paper, we revisit most of the aspects of NoC design beginning from microarchitecture and design methodologies and moving to physical integration challenges (clocking strategies, runtime adaptivity), as well as network partitioning, reconfiguration and virtualization and identify the challenges involved in future NoC design. At the same time, the adoption of emerging interconnect technologies such as optical and wireless interconnects are thoroughly discussed and their associated challenges in system architecture, circuit design, device fabrication and CAD tool development are analyzed. The selected research avenues are also supported by an industrial viewpoint that discusses the NoC design arena of current and future SoCs for mobile platforms.

2 NoC microarchitecture challenges

All aspects of Network-on-Chip architecture starting from topology and routing algorithms [7–11], and covering router and network interface microarchitecture have evolved significantly over the last decade. Although topologies and routing algorithms are defined in a generic manner [12,13], allowing several design customization and specialization decisions, the same does not hold for router microarchitecture.

A unified customizable model that will cover in a unified manner all micro-architectural alternatives such as control and data path pipelines [14–17], speculation [18,19], buffering architecture [20–23] and allocation policies [24,25] is still missing. The derivation of such a model would allow for rapid, safe architectural changes in order to find the globally optimum architectures bridging the gap between architecture exploration, microarchitecture fine tuning and physical implementation. The derived customizable model should be smart enough to differentiate its microarchitecture depending on switch radix, centralized or distributed physical placement [26,27] and available silicon area.

Currently, router design involves assembling hardware blocks from a component library of varying granularity and complexity. Although such an approach has provided so far efficient architectures, its efficiency is limited by the efficiency of the independent blocks; the designer's potential for delivering efficient compositions, and the depth of the design space exploration. At the same time, every new synthesis of components would require separate verification effort that would first guarantee correctness such as deadlock and starvation avoidance, performance validation and separate physical implementation characterization and integration with the remaining components of the system. A predefined customizable model would significantly reduce verification and validation effort since its generic definition and pre-characterization is expected to cover by construction every customized instance of it.

At the same time, on-chip network interfaces that decouple at the eye of the programmer computation from communication have evolved in recent years from primitive protocol bridges and packetizing-buffering modules to sophisticated NoC units. The network interfaces besides physical integration including clock domain synchronization and power domain interfacing are responsible for many other higher-level-abstraction structures close to the programming model and to the communication protocol stack. The architecture of network interfaces does not follow a standard approach and many ad hoc alternatives supporting various application domain features have appeared recently [28–31].

The trend in network interface design is to integrate as many networking features as possible directly in hardware while keeping a balance between hardware complexity of the interfaces in terms of area and latency relative to the connected cores and the complexity of the rest of the network. The architectural features that every network interface should support include: (a) protocol and bandwidth adaptivity (b) buffering, (c) error management, (d) quality of service, (d) memory address protection and isolation (security), (e) out of order transactions handling, and (f) programming interface support [32] that should follow a standard form such as MCAPI or any other generic interface that would appear in the future.

Although there is a growing consensus on what a network interface should support the same does not happen on how it should be implemented. A clear *fixed* architecture and possibly programmable by a small instruction set architecture is missing that would allow easy customizations and versatile operation via software-like reprogrammability. Also part of the communication libraries and application-level abstractions could be implemented easily in software via the network interface's instruction set.

Besides improving NoC architectures either at the nodes of the network or at the interfaces, holistic design methodologies should also evolve possibly moving to a true hardware-software interconnect co-design. A clear asset in this direction would be the definition of interconnect-specific architecture description languages and automatic and correct-by-construction model to RTL synthesis methodologies (like a network-specific high-level synthesis methodology [33,34]) that would allow for faster exploration of networking architecture and holds promise for most efficient design by allowing joint and across layer optimizations [35]. Up to now interconnect design followed a layered approach that encapsulates networking functions in a hierarchical stack of per-layer operations (e.g. link, transport, network layer). Designers focus on a particular layer that hinders possible across-layer optimizations and maintains unnecessary overhead hidden inside each layer. Although, concrete studies are still missing on how the network *intelligence* should be split between interfaces and network nodes, the definition of a network-centric design methodology at the architecture level would allow new unexplored alternatives.

Interconnect-centric design technologies need to be supported by new efficient usage strategies of the network as a whole. Traditional network usage deals with bandwidth and latency guarantees trying to satisfy either hard communication deadlines or offer equality of service [36]. Although such techniques have not reached the highest plateau of efficiency or ease of implementation new usage strategies should evolve that consider energy efficiency, energy fairness, and packet-delivery criticality in a unified manner.

The wide adoption of NoC technology in today's high-end SoCs is expected to grow the next years even to ultra low power SoC platforms that are gradually becoming multicores. In such systems, maximum energy efficiency is the most important quality factor and thus operate at low voltages (and possibly at low speeds) trying to keep power consumption close to the minimum possible. The application of NoC technology, as we know it today, in such platforms would require a complete redesign from the circuit up to architectural level of all traditional NoC components (buffers, crossbar, arbiters) and topologies making them truly voltage scalable and adaptive to changing operating conditions [37]. Low cost on-line fault tolerance features may help in this direction when applied appropriately so as not to eat back of the energy saved by voltage scaling.

3 The compositional challenge

Power dissipation continues to be a primary design constraint in the multi- and many-core chip era. Increasing power consumption not only results in increasing energy costs, but also results in high die temperatures that affect chip reliability, performance, and packaging cost. From the performance standpoint, current and future largely integrated systems will have to carefully constrain application performance to stay within power envelopes [38]. Fortunately, multi-core systems host applications that exhibit runtime variability in their performance requirements, which can be exploited to optimize throughput while staying within the system power envelope.

Dynamic voltage and frequency scaling (DVFS) schemes seek to exploit runtime variability in application behavior to achieve maximum energy savings with minimal performance degradation. The granularity of adaptive voltage and frequency control is currently still an open issue. The milestone Intel Single-Chip Cloud Computer implementation exhibits 24 frequency islands with 15 speed settings from 100 to 800 MHz, and 7 V islands with 7 V levels from 0.7 to 1.3 V in steps of 0.1 V [39]. The on-chip network is a voltage and frequency island on its own. Considering that the chip consists of 48 cores structured into 24 tiles, it can be considered a relevant example of fine-grained power management enabled by on-package voltage regulators. In practice, even though the performance advantages of per-core DVFS in multi-core systems have been suggested [40,41], providing per-core, independent voltage control can be overly expensive [40]. In contrast, when DVFS is applied across multiple cores, determining a single optimal DVFS setting that simultaneously satisfies all cores will be extremely difficult; some applications will suffer performance loss or power overheads. This problem worsens as the number of cores and running applications increase in future systems. There is currently no generalized consensus on the above tradeoffs, however the relentless improvement of (integrated) voltage regulator technology as a standard homogeneous CMOS component holds promise of proliferating DVFS domains for maximum energy benefits [42]. As a consequence, a future scenario can be reasonably envisioned where the homogeneous cores of a regular tile-based architecture will deliver heterogeneous power-performance operating conditions with fine-grained granularity, and with runtime tuning capability.

Last but not least, future many-core programmable accelerators are likely to host a large number of concurrent applications to enable effective exploitation of the hardware resources, thus pushing the partitioning concept [43]. The virtualization paradigm, which is becoming mainstream also in the embedded computing domain, is fostering this trend, since virtual machines might be easily allocated a subset (possibly changing over time) of the parallel computer architecture, with trustworthy isolation strategies [44]. Overall, at any given point in time portions of the many-core architecture might go unused, hence they could be effectively powered off [45].

The above requirements have relevant implications on hardware design. Especially, they cause design-time homogeneous architectures to become highly heterogeneous at runtime, given the diversity of core power states across the platform. This diversity should be absorbed at component boundary, where IP cores are interconnected with the system integration framework, that is with the on-chip communication architecture. Networks-on-Chip should be therefore ready to deliver communication paths at runtime that potentially cross areas with highly heterogeneous operating conditions.

Synchronizer-based design is the traditional answer to this challenge [46], since it enables to absorb clock phase and frequency setting differences across communicating frequency islands at some synchronization overhead, especially latency and power. Unfortunately, the latency overhead caused in NoC links in turn calls for larger buffering requirements at the receiver end, in order to preserve the capability for maximum throughput operation [47]. Synchronizers are extensively used in embedded systems, especially dual-clock FIFOs (like for instance [48], however they currently cope only with coarse-grained splitting into frequency domains in most cases. As finer-grain splitting will gain momentum, optimization techniques for them will become mandatory to fit the tight resource budgets [49]. The literature is currently ready to deliver novel proposals and interesting ideas, although architectural optimizations and silicon validations are far from complete [50]. As an example, a merging technology of synchronizers with NoC building blocks was proved capable of reusing expensive buffering resources for multiple purposes, including synchronization, performance buffering and flow control [51]. Nonetheless, the distinctive challenges for the industrialization of synchronizer-based technology encompass:

- The careful engineering of timing constraints across links, since the need to deliver flow control in NoCs causes non-trivial round-trip channel dependencies even in source-synchronous communication.
- The continued development of bundled data routing methodologies in sub-40 nm technologies, capable of keeping relative delay mismatches between link wires under control.
- The implementation of clock gating techniques capable of cutting down on idle power, especially when source-synchronous clock signals need to be routed to the receiver end for the sake of signal resynchronization.
- The implementation of reliable reset mechanisms, safeguarding operation of those synchronizers that require precise alignment between their front-ends (in one clock domain) and their back-ends (in another clock domain) with respect to timing uncertainties in reset deassertion across clock domains.
- The proper (over)sizing of the number of cascaded stages in brute-force synchronizers to counter the degradation of the resolution time constant of synchronizers as technology scales deeper into the nanoscale regime. This has non-negligible implications over the performance and power overhead of dual-clock FIFOs, which would in many cases require slot overprovisioning to preserve the full throughput operation capability.

There is no doubt that reliable and energy-efficient many-core system design will only be feasible under relaxed synchronization assumptions in the future. An indirect confirmation comes from the analogy of the synchronization paradigm used in two relevant Intel test chips. On one hand, the early 80-core Intel Polaris chip relied on mesochronous clocking to implement simpler, and less power-hungry clock distribution networks replacing a complicated H-tree with something simpler and shorter like a grid [52]. On the other hand, the latest demonstration of a 256-node Intel NoC on 22 nm Tri-Gate CMOS relies on the principle of source-synchronous communication, which is effectively coupled with an hybrid packet/circuit switching architecture built on top of it, thus yielding 20.2 Tbps among the nodes and 18.2 Tbps/W efficiency when running at 430 mV in near-threshold voltage operation [53].

An appealing alternative to synchronizer-based design consists of clockless handshaking [54]. When applied for inter-domain communication, it holds promise of average-case instead of worst-case performance, no switching power of a clock tree, especially in idle state, robustness to process/voltage/temperature variations, and efficient delivery of differentiated per-link performance [55]. Counterintuitively, such potential benefits are not reflected in an adequate industrial exploitation, which includes asynchronous Ethernet routers and high-speed FPGAs, and only marginally on-chip interconnect sub-systems.

The traditional explanation consists of the poor CAD tool support to design asynchronous systems. Many efforts are underway to get predictable and fast-converging designs by means of ad-hoc tools [56] and/or scripting and methodologies on top of mainstream CAD tools [57], however they are currently not capable yet to avoid extensive manual intervention, the deactivation of relevant optimization capabilities of such tools [58], the technology-specific description of some components in abstract specifications, and to enable the flexibility required for the design and synthesis of soft macros. As a consequence, prototype designs and real products still largely rely on hard macros and full custom design when it comes to the asynchronous components [59], with some noticeable exceptions [60,61].

However, it should be observed that existing industrial prototypes do not often represent an incentive for further development of the tooling support. Existing asynchronous interconnect fabrics can easily prove relevant savings on application total power, however they feature a larger energy-per-flit than their synchronous counterparts. As a result, power savings can be explained only in terms of the poor utilization of the system interconnect, since the idle power figure is clearly in favour of the asynchronous implementation. The ultimate reason for this trend is that asynchronous NoCs are typically designed with 4-phase communication protocols and delay-insensitive data encoding. The former choice implies two complete round-trip channel communications per transaction, which becomes unaffordable in the presence of long links. The second choice guarantees high timing robustness since circuit functionality and operation are guaranteed by construction in the face of delay variations during the fabrication process. Unfortunately, this comes with high area occupancy, low coding density and high energy per bit.

More recently, these quality metrics are raising the interest for an alternative design style relying on bundled data [62–64]. The key rationale is that if the lower timing robustness of this data encoding can be kept under control by means of efficient CAD tools and guardbands on relative timing constraints, the benefits of reduced area, reduced wire-per-link and reduced energy-per-bit can be materialized. The success of this design style will probably depend on the availability of efficient tunable delay lines to make the performance penalty of the guardbands link-specific. Whether this is a viable solution for high-performance designs or not, bundled data clearly represents the way to go for the desynchronization of low-power, low-end designs in the embedded computing domain. One possible obstacle is given by the fact that latch performance might not be that good in low-power technology libraries, which questions the typical design practice of delivering high-performance asynchronous designs via extensive utilization of pipelining techniques in the presence of specific pipeline design styles (e.g., MOUSETRAP pipelines).

Overall, the question about the role of asynchronous interconnect technology for the solution of the compositional challenge in manycore systems is far from getting stable answers. The key driver is the possibility to connect domains regardless of their specific and runtime-varying operating conditions, like in [65,66]. Literature keeps documenting remarkable and trustworthy power savings whenever the technology is applied. It should be however brought to the stage where its development for exploitation within an industry-standard methodology and tool-flow becomes cost-effective with respect to the further evolution of current design methods.

4 The resource sharing challenge

From previous sections we have seen many challenges ahead to be addressed for proper NoC design. However, the fact that the NoC is a shared resource within the chip makes its design much more critical than expected. If we think of a possible multicore system, where tens or hundreds of processor cores communicate within them and with caches and memory controllers, we easily see that the medium used for that communication is always the same, the NoC. Thus, the way we design the NoC will heavily influence not only the overall system performance but the way we share resources (cores, caches, memory controllers, accelerators, …). This imposes an orthogonal challenge to the previous ones since an unbalanced use or resources will lead to poor performance numbers, or even to unattainable QoS levels.

Indeed, future systems with hundreds, or perhaps thousands of cores, will inherit a structural problem. Applications running on such systems will simply not scale, or will scale poorly, thus not taking benefit of the theoretical peak performance of those systems. To address this issue, we can see high-performance computing systems divided in two categories [67]. In *capability computing*, HPC infrastructures (supercomputers) are used to solve a single and highly complex problem in the shortest possible amount of time. In *capacity computing*, however, a compute system solves as many problems as possible in parallel with the lowest possible cost. This refers to, for instance, data-centers receiving millions of requests per time unit.

If we apply the *capacity computing* approach to multicore systems, we can think of applications (or tasks) running on the same system but using disjoint sets of resources (cores, memories, …). This is indeed an appealing approach, since it allows to maximize system resources utilization, thus making a proper use of our system. This approach (sharing resources between different applications or tasks) is also emerging in the embedded domain with the concept of mixed-criticality systems (MCS) and virtualization. In MCS systems, a single multicore chip must be able to run different applications with different criticalities and must guarantee failures or perturbations of applications do not affect the other applications performance. Virtualization of chip resources is also being promoted lately and also imposes an effective resource sharing policy in order to decouple application's performance from the rest. In the near future, multicores will invade every domain (e.g. aerospace, automotive) and will demand for efficient policies to manage the resources in a structured and safe manner. Unfortunately, the NoC is a shared resource and lays in the middle of the problem. So, it is clear we need to design the NoC with these new requirements in mind.

This challenge is aggravated by the fact that we may have running on top of the system a coherence protocol, which will guarantee memory access consistency. In such scenario the NoC will face mainly the traffic generated by the coherence protocol. This kind of traffic has its own characteristics, such as traffic distribution, traffic burstiness, and communication types (unicast, collective, gather operations). An NoC can not be designed without taking into account such traffic characteristics.

Support of efficient sharing resource policies and coherence protocols demands for efficient NoC designs in the following directions. First, the communication particularities of coherence protocols need to be supported natively by the NoC. Examples are broadcast support in the NoC allowing efficient communication of coherence protocol commands, gathering operations support in the NoC allowing efficient acknowledgments of multiple cores to the same memory block, and efficient support of synchronization primitives in the NoC allowing fast synchronization operations of the processes running on the system. These kind of optimizations (indeed, an efficient co-design of the NoC and the coherence protocols) will allow such protocols to scale, or at least scale better, thus delivering higher capacities and performance numbers. Second, the NoC needs to be designed with built-in mechanisms and methods to guarantee runtime and flexible partitioning schemes, which will enable effective isolation of applications or tasks in the same chip. This affects mainly to the design and properties of routing algorithms. Topology-agnostic routing algorithms (like up*/down* or segment-based routing) allow the building of partitions with any shape, thus promoting the partitioning capability. Third, the NoC needs to be designed with reconfiguration capabilities. Indeed, if we plan to map different applications on the system these applications will be continuously entering and leaving the system, thus needing different numbers and types of resources and demanding a proper chip reconfiguration. An NoC with transparent reconfiguration (not affecting the current traffic and not stopping the traffic) is required for the support of such systems. Finally, the final direction to take is a complete and transparent exposure of the configuration and partitioning capabilities of the NoC to the software stack, mainly the operating system and in particular the hypervisor. This module will be in charge of customizing the system to the current demands of the applications requesting service from the system.

All this support has its center of gravity in the NoC since is the shared resource and is the one used to communicate all the system components within the chip. Proper design of the NoC to support partitioning while optimizing coherence protocols support will become mandatory for future chips based on NoCs.

5 Emerging interconnect technologies

According to the ITRS roadmap [68], interconnect innovation is the key to satisfying performance, reliability, and power requirements in the long term. Future interconnect technologies must support ultra-high data rates (e.g., greater than 100 Gbps/pin), be scalable enough to support tens to hundreds of concurrent communication streams, and involve fabrication techniques that are compatible with mainstream MPSoC and system-in-package (SiP) technologies. An overview of the fundamentals and ongoing research challenges for two revolutionary interconnect technologies is reported in [69,70], namely silicon nanophotonics and RF/wireless interconnects.

Optical links are already pervasive in data centers because of their ability to improve the bandwidth density over copper cables, to the point that optical switching represents the next step in order to overcome the overhead of frequent domain conversions [71]. There is instead no consensus on the use of silicon photonics for on-chip communication, where optimized eletronic links, and their evolutions, are competitive. Yet, emerging nanophotonic technology has yielded a rich design space for on-chip optical-electrical architectures [72–76]. Early studies such as those in [72,74,75,77–80] made the point for the performance and power properties of photonic interconnection networks in isolation from the rest of the system. System-scale analysis was instead made affordable by [81], with a trade-off between accuracy and simulation speed in favour of the latter. The need to come up with compelling cases for silicon nanophotonic technology has motivated the quest for higher accuracy, for instance by considering communication workloads [82], or the network interface overhead [83]. As the level of detail in comparative analysis between electrical and optical fabrics increases, it is becoming evident that while the optical interconnect fabric is not more energy efficient per se [84], the optically-augmented system is, since it can burn power for a lower amount of time due to the lower execution times that optical links enable [85].

Another alternative for future on-chip communication consists of NoCs with multi-band RF interconnects (RF-I) [86]. In this particular NoC, instead of depending on the charging/discharging of wires for sending data, electromagnetic (EM) waves are guided along on-chip transmission lines created by multiple layers of metal and dielectric stack. As the EM waves travel at the effective speed of light, low latency and high bandwidth communication can be achieved. Though RF-I NoCs can be built using existing CMOS technology, they require laying of long on-chip transmission lines to serve as wave guides, without eliminating any existing links.

Recently, the design of a wireless NoC based on CMOS Ultra Wideband (UWB) technology was proposed [87]. In [88], the feasibility of designing on-chip wireless communication networks with miniature antennas and simple transceivers that operate at the sub-THz range of 100–500 GHz has been demonstrated. If the transmission frequencies can be increased to THz/optical range then the corresponding antenna sizes decrease, occupying much less chip real estate. One possibility is to use nanoscale antennas based on CNTs operating in the THz/optical frequency range [89]. Consequently, building an on-chip wireless interconnection network (WiNoC) using THz frequencies for inter-core communications becomes feasible.

On-chip wireless communication links not only alleviate the latency and energy dissipation issues of conventional technologies but also eliminate complex interconnect routing and layout problems arising in some of the alternative technologies. Hence, such interconnects enable design of novel and efficient architectures which mitigate the multi-hop communication of traditional NoCs to achieve significant performance gains. A detailed survey regarding the promises and design challenges of this emerging paradigm is reported in [90].

The development of emerging interconnect technologies implies that three fundamental gaps need to be addressed by researchers at different levels of abstraction, and with a cross-layer approach to design and optimization. Among them, the *physical design gap* is certainly the most evident issue.

As regards optical NoCs, high-speed, low power, and small feature-size electro-optical modulators and photo-detector receivers need to be developed, since their quality metrics, together with the overhead of laser sources, will determine the threshold required to be advantageous over electrical interconnects. In particular, high-speed, electrically-driven monolithic light sources have remained elusive so far, thus calling for profound innovations in the field of integrated on-chip light sources. Finally, on-chip optical interconnect modules are very sensitive to process and thermal variations. Designers need to ensure active or passive optical control methods to maintain reliable device operation [91].

Similarly, the effectiveness of WiNoCs strongly depends on the design of the physical layer. In turn, the miniaturized on-chip antennas and the wireless transceivers influence the performance of the physical layer. Characteristics of the antennas and the transceivers also depend on the adopted frequency range of communication (ultra wide band, millimeter-wave, sub terahertz, or terahertz). All physical layers designed in different frequency bands have antenna and transceiver area and power overheads [90]. Thus, innovations such as [92–94] are required to achieve the best performance-overhead trade-off and fully exploit the advantages of wireless links.

Technology maturity is not the only gap that separates emerging technologies from their industrial uptake. An *architectural gap* in fact raises on top of the physical one, although they end up being tightly intertwined.

For optical NoCs, building a communication architecture out of a specific optical toolbox is a complex task that spans several design concerns. After all, from a functional viewpoint an optical network is nothing else but a non-blocking crossbar, due to the lack of buffering technology of practical relevance. Hence, the control complexity is entirely moved to the boundaries. There, key aspects such as flow control, synchronization, buffering architecture, resource sharing techniques, serialization, etc. should be taken care of, and may determine the threshold beyond which an optical interconnect is better than another one, or than an electrical counterpart. A key architecture-level design decision concerns the implementation of space-routed optical NoCs, which devote the available WDM (wavelength-division multiplexing) link bandwidth to peer-to-peer communications, or wavelength-routed NoCs, which exploit the same bandwidth for the sake of delivering global, contention-free communication, while decreasing the available bandwidth for the specific communication flows. This decision tightly depends on the application requirements, since wavelength-routed NoCs are well suited for latency-critical applications, while space-routed ones are the best choice for throughput-intensive applications, especially for long-lasting connections. As pointed out in [95], there are a number of intermediate solutions between the two extreme cases, which yield to photonic bus variants. With the single writer multiple reader paradigm (SWMR), additional signaling is required for the sake of tuning the filters of the intended receiver, while in the multiple writer single reader (MWSR) paradigm a global arbitration is needed to select the injecting sender into the photonic bus. The multiple reader multiple writer scenario is also feasible. The chosen scheme has then serialization and scalability implications, that are especially constraining for wavelength-routed networks due to the large amount of needed resources. Currently, an extensive comparison of architectural solutions in an homogeneous experimental setting is still missing, and so is a study relating them to the requirements of realistic workloads.

Clock resynchronization is another architecture-level concern. Some parts of the optical network interface need in fact to work at overly high speed (e.g., 10 GHz, associated with the modulation rate of the optical medium, and with the serialization ratio), or with multi-phase clock signals. This is not only a physical design issue, since signals converted back from the optical to the electronic domains should be resynchronized in the target clock domain [83]. The resynchronization architecture and circuitry is still a largely unresolved issue for optical NoCs, although source synchronous schemes seem to be the preferred option. Unfortunately, they require the transmission of clock signals across the optical domain, which becomes therefore an active research field [96].

Addressing the design predictability gap is mandatory when selecting the target topology for the optical NoC. Although not explicitely stated, logic topologies are tied to implicit placement constraints for initiator and target interfaces. The real positioning of such interfaces on the layout of the system at hand may cause a radical change of the physical routing paths. Side effects are an increased length of the waveguides or an unexpected number of additional waveguide crossings. Placement constraints are especially severe in a 3D stacked environment, as proven in [97], thus justifying their consideration upfront in the design process [98].

WiNoC architectures can be assembled by overlaying a regular wired mesh-based NoC with wireless links [93,99]. However, there is currently an immense interest in creating novel architectures aided by the on-chip wireless communication [94]. In this direction, different hierarchical small-world wireless NoC architectures incorporating THz and mm-wave wireless links are explored in [94] and [100], respectively. These works have demonstrated that, by using wireless links as long-range communication channels between widely separated cores along with wired interconnects connecting adjacent cores, it is possible to obtain significant gains in achievable bandwidth, and improve the energy dissipation profiles without introducing significant hardware overhead. Research in this domain is far from being consolidated.

Moreover, to attain the desired performance benefits using WiNoC, the available communication resources should be utilized optimally. Therefore, efficient media access mechanism [87,93,100,101], along with optimum routing protocol [87,93,94,102], is crucial for efficient utilization of the wireless channels. Since all solutions improve the achievable bandwidth at an area and power overhead, a comprehensive study quantifying merits and limitations of these techniques, and their implementation challenges, needs to be carried out for an informative comparative analysis.

The MAC and routing protocols for WiNoCs need to be complemented by suitable flow control mechanisms to enable optimum utilization of the wireless medium [103,104].

Last but not least, challenges in reliability and integration demand radically different architectural design to make this emerging interconnect paradigm viable for large-scale adoption. Although architectural innovations such as [105] may enable resilience against permanent failures, the wireless channels are inherently more prone to transient errors than their wireline counterparts. In this direction, it is demonstrated in [106] that with carefully designed error control coding (ECC) schemes in the WiNoC it is possible to achieve high gains in performance due to the wireless links while maintaining reliability comparable to that of a traditional wire line NoC. However, application of ECC also introduces timing and area overhead, which gives rise to an interesting trade-off to explore.

In addition to the physical and architectural gaps, the system as a whole should be optimized around an optical transport medium, which implies the codesign of components together to meet system-level requirements (i.e., *the systemability gap*). This includes for instance the codesign of the fabric with the cache coherence protocol, the routing path selection policy in case of hybrid interconnect fabrics, the differentiated service of latency- vs. throughput-critical traffic, the codesign of the on-chip network with the processor-memory (including off-chip) network, the avoidance of message-dependent deadlock. At this level, also compiler and software optimizations should be considered, including for instance the optimization of the dynamic behaviour of the application and the exploitation at runtime of the available degrees of freedom in the communication fabric.

Finally, each new technology should come with its own design technology support from the ground up, in order to bridge the gap between physical designers and system designers, who need to do design with the new technology. This encompasses abstract models, design methodologies, tools and toolflows. For instance, abstraction layers and associated description tools should be redefined for optical NoCs, thus matching the electronic definitions of behavioural views, RTL ones, etc. Moreover, new tools for placement and routing such as [107] are needed, due to the inherently different optimization metrics that optical NoCs require with respect to mainstream CAD tools for electronic design (e.g., number of waveguide crossings, waveguide length, or both).

Overall, the most daunting challenge for the next few years will be to come up with compelling cases for silicon nanophotonic as well as for wireless networks, thus possibly justifying the definition of roadmaps and investments in technology development based on solid experimental evidence. Assessing the implications of an emerging interconnect technology over the quality metrics of real-life devices such as GPUs or programmable accelerators is part of this needed validation framework. In this respect, the experience of researchers starts to put together a few basic rules for trustworthy crossbenchmarking between NoCs on top of emerging technologies vs. their electrical counterparts. Next, such rules are reported by deriving them from the converging conclusions of [95] and [84]. They are tailored to optical interconnection networks, although the inspiring principles behind them could easily drive WiNoC research as well in the future:

- **clearly specify the logic topology** In many papers, logic topologies are hardwired with their physical implementations, hence preventing a true distinction of design points, and the application of well-known optimization principles from interconnection network theory.
- **explore the space of mapping options to nanophotonic devices** For a given logic topology, different technology mappings do exist, characterized by the use of a different mix of photonic devices (e.g., 1x2, 2x2 or higher-order photonic switching elements), or a different filtering order of WDM signals.
- **account for place&route constraints** The actual gap between logic topologies and their physical implementation under the place&route constraints of the layout at hand should be quantified. Also, design techniques/choices should be investigated/made to minimize such a gap.
- **keep it simple** Simple interconnection solutions, starting from topology selection and from the choice of basic building blocks, are a must in order to minimize the adoption risk of a new technology.
- **design the network interface architecture** Interfacing the electrical and the optical domains is not just an issue of bringing optoeletronic devices in the design, but it is an architecture-level effort too, where networking design issues (buffering, flow control, deadlock avoidance, etc.) should be addressed.
- **use an aggressive electrical counterpart** Previous work often reports orders of magnitude better performance and power of the optical fabric also because the electrical counterpart is built on top of naive assumptions. A trustworthy crossbenchmarking should consider state-of-the-art electrical NoC architectures, which should undergo synthesis on top of industrial technology libraries.
- **assume a broad range of device parameters** In the presence of a fast evolving technology, it does not make sense to tie conclusions to specific parameters for the silicon photonic devices. Rather, parametric studies should define the requirements for physical designers in order for their devices to be mature enough for practical exploitation.
- **carefully consider static power overhead** Optical interconnect technology is static-power dominated, while materializing excellent dynamic power savings. Therefore, previous studies suggested not to use it for short-range communications, but rather to aggregate injecting cores into optical network interfaces, while performing short-range communications still in electronics. This avoids the proliferation of domain converters, which would consolidate the static power dominance.

Obviously, the above rules should be followed not only to determine under which operating conditions switching to a new interconnect technology is to be preferred to the further evolution of current electrical links, but also to compare emerging technologies with one another, thus complementing seminal works in this field [108].

6 Particular challenges for mobile platforms

The heart of mobile platforms like tablets and mobile phones are heterogeneous MPSoCs. They do not simply comprise a number of identical processing nodes as mobile platforms run on batteries and all processing and communication tasks need to be executed very efficiently. An increasingly large number of specialized hardware accelerators is supporting the general-purpose processors in order to achieve a high efficiency, i.e. executing the desired operations and tasks faster and/or more energy efficient than the main CPUs could do. Furthermore, these platforms comprise a zoo of different specialized components such as display controllers, camera interfaces, sensors, connectivity modules such as Bluetooth, WiFi, FM radio, GNSS (Global Navigation Satellite System), and multimedia subsystems. Most of them provide local intelligence such as an integrated processing core or a DSP. All of these subsystems and components communicate with each other and with memories such as on-chip SRAMs, and SDRAMs. A sophisticated on-chip interconnection network is required which needs to be tailored to the communication requirements of each individual component. Conventional 2D mesh networks or simple rings are not appropriate here. In fact, these networks need to be as heterogeneous as the rest of the SoC. This complicates both architecture and topology decisions, and poses many practical challenges on top of typical challenges from academia such as minimum hop-count or deadlock avoidance.

Battery life time and responsiveness define the user experience. Both things at the same time can only be achieved by highly optimized architectures. This implies extensive power saving features such as DVFS and scenario-based sleep modes for parts of the system. Based on the given scenario, portions of the system are put into low-power modes. This includes reduced frequencies, lower voltages or the complete power down of a subsystem. The chosen power-saving measure depends on the wake-up time that is needed to restore the subsystem to its previous state. Dynamic clock frequency scaling can be used easily as the wake-up time is in a range of few clock cycles. The subsystem is still able to work, although slower, and will retain its state. A complete power-down of a subsystem, however, is only initiated if the power-down phase is long enough compared to the phases of powering down and waking up. In some cases the start of the next active phase cannot be determined in advance. It could happen that the subsystem is requested to be active immediately after initiating the power-down procedure which is then immediately followed by the wake-up procedure. Both procedures might use more energy than keeping the subsystem in active (idle) state. So, the power-down phase must be long enough to pay off from both timing and energy-saving perspectives.

From a physical design point of view the increasing complexity of the computing and communication architectures must be handled properly. Clock distribution and synchronization inside the chip become prohibitively costly. Long clock signal wires running across the chip and toggling at high frequencies are difficult to balance and consume lots of energy. Mesochronous and asynchronous islands arise to overcome the need of long clock wires. However, these islands have to be integrated seamlessly. While this sounds trivial to accomplish, in practice many things need to be taken into account. Re-synchronization between clock domains needs to be performed, which causes additional latency and might even cause throughput reduction on control and data paths. A sophisticated design flow is needed, but also architectural awareness in order to do more good than bad. Typically, these data and control paths will be hand-optimized by experienced engineers to squeeze the last drop of performance out of these connections.

The next step in mobility is the Internet of Things (IoT) [109] where everyday devices, objects and physical assets are equipped with computing power, sensors and wireless connections that enable them to communicate with each other and with the Internet. These connected objects become more and more intelligent with tiny integrated processors such as the Intel Quark processor, or even be complete computers such as the Intel Edison, a PC in the size of an SD card including Bluetooth and WiFi communication capabilities. In the end, these SoCs with their on-chip interconnects will be connected forming large wireless off-chip networks with an ever increasing amount of data that will be transmitted.

References

1. Handy J (2011) NoC interconnect improves SoC economics. Objective analysis—semiconductor market research
2. Browne J (2012) On-Chip Communications Network. in Sonics
3. Dally WJ, Towles B (2001) Route packets, not wires: on-chip interconnection networks. In: Proceedings of the 38th design automation conference (DAC)
4. Wentzlaff D et al (2007) On-chip interconnection architecture of the tile processor. IEEE Micro, pp 15–31
5. Benini L, De Micheli G (2002) Networks on chips: a new SoC paradigm. IEEE Comput 35(1):7078
6. De Micheli G, Seiculescu C, Murali S, Benini L and Angiolini F et al (2010) Networks on chips: from research to products. In: 47th design automation conference (DAC 2010)
7. Kim J, Balfour J, Dally WJ (2007) Flattened butterfly topology for on-chip networks. In: Proceedings of IEEE/ACM international symposium on microarchitecture (MICRO)
8. Mishra AK, Vijaykrishnan N, Das CR (2011) A case for heterogeneous on-chip interconnects for cmps. In: Proceedings of the international symposium on computer, architecture, pp 389–400
9. Flich J, Mejia A, Lopez P, Duato J (2007) Region-based routing: An efcient routing mechanism to tackle unreliable hardware in networks on chip. In: International symposium on networks on chip (NOCS)
10. Ma S, Enright Jerger N, Wang Z (2012) Whole packet forwarding: efficient design of fully adaptive routing algorithms for networks-on-chip. In: Proceedigs of the international symposium on high performance computer, architecture, pp 467–478
11. Seo D, Ali A, Lim W-T, Rafique N, Thottethodi M (2005) Near-optimal worst-case throughput routing for two-dimensional mesh networks. In: Proceedings of the 32nd annual international symposium on computer architecture (ISCA '05). IEEE Computer Society, Washington, DC, pp 432–443
12. Balfour J, Dally WJ (2006) Design tradeoffs for tiled CMP on-chip networks. In: Proceedings of the 20th ACM international conference on supercomputing (ICS)
13. Passas G, Katevenis M, Pnevmatikatos D (2012) Crossbar NoCs are scalable beyond 100 nodes. IEEE transactions on computer-aided design of integrated circuits and systems (TCAD). 31(4):573–585. ISSN: 0278–0070
14. Azimi M, Dai D, Mejia A, Park D, Saharoy R, Vaidya AS (2009) Flexible and adaptive on-chip interconnect for tera-scale architectures. Intel Technol J 13(4):6277
15. Kim J (2009) Low-cost router microarchitecture for on-chip networks. In: International symposiun on microarchitecture
16. Salihundam P et al (2010) A 2Tb/s 6x4 mesh network with DVFS and 2.3Tb/s/W router in 45nm CMOS. In: Sympsosium VLSI circuits
17. Vangal SR et al (Jan. 2008) An 80-tile sub-100-W teraFLOPS processor in 65-nm CMOS. IEEE J Solid-State Circuits 43:6–20
18. Peh L-S, Dally WJ (2001) A delay model and speculative architecture for pipelined routers. In: Proceedigs of the 7th international symposium on high-performance computer, architecture (HPCA-7)
19. Mullins RD, West AF, Moore SW (2004) Low-latency virtual-channel routers for on-chip networks. In: Procedings of the international symposium on computer, architecture, pp 188–197
20. Tran AT, Baas BM (2011) RoShaQ: high-performance on-chip router with shared queues. In: IEEE ICCD, pp 232–238
21. Becker DU (2012) Adaptive backpressure: efficient buffer management for on-chip networks. In: IEEE ICCD
22. Hassan SM and Yalamanchili S (2013) Centralized buffer router: a low latency, low power router for high radix nocs. In: IEEE/ACM international symposium on network on chip
23. Seitanidis I, Psarras A, Dimitrakopoulos G, Nicopoulos C (2014) Elastistore: an elastic buffer architecture for network-on-chip routers. In: Proceedings of design automation and test in Europe (DATE)

24. Michelogiannakis G, Jiang N, Becker D, Dally W.J (2011) Packet chaining: efficient single-cycle allocation for on-chip networks. In: Proceedings IEEE/ACM international symposium on microarchitecture (MICRO), pp 83–94
25. Dimitrakopoulos G et al (2013) Merged switch allocation and traversal in network-on-chip switches. In: IEEE transation on computers
26. Roca A, Hernandez C, Flich J, Silla F, Duato J (2013) Silicon-aware distributed switch architecture for on-chip networks. J Syst Archit 59(7):505–515
27. Balkan A, Qu G, Vishkin U (Oct 2009) Mesh-of-trees and alternative interconnection networks for single-chip parallelism. IEEE Trans VLSI Syst 17(10):1419–1432
28. Saponara S, Bacchillone T, Petri E, Fanucci L, Locatelli R, Coppola M Design of a NoC interface macrocell with hardware support of advanced networking functionalities. In: IEEE transactions on computers
29. Yang X, Qing-li Z, Fang-fa F, Ming-yan Y, Cheng L (2007) NISAR: an AXI compliant on-chip NI architecture offering transaction reordering processing. In: Proceedings of the 7th international conference ASIC ASICON 07, p 890893
30. Radulescu A, Dielissen J, Pestana SG, Gangwal OP, Rijpkema E, Wielage P, Goossens K (2005) An efficient on-chip NI offering guaranteed services, shared-memory abstraction, and flexible network configuration. IEEE Trans Comput Aided Design Integr Circuits Syst 24(1):417
31. Ebrahimi M, Daneshtalab M, Liljeberg P, Plosila J, Tenhunen H (2010) A high-performance network interface architecture for NoCs using reorder buffer sharing. In: Proceedings of the 18th Euromicro international parallel, distributed and network-based processing (PDP) conference, p 546550
32. Kavadias S, Katevenis M, Pnevmatikatos D (2011) Network interface design for explicit communication in chip multiprocessors, chapter 10. In: Flich J, Bertozzi D (eds) designing network-on-chip architectures in the nanoscale era. CRC Press–Taylor & Francis Groupa, pp 325–351. ISBN: 978-1-4398-3710-8
33. Fingeroff M (2010) High-level synthesis blue book. Xlibris Corp
34. Coussy P, Morawiec A (2008) High-level synthesis: from algorithm to digital circuit. Springer
35. Shacham O, Azizi O, Wachs M, Richardson S, Horowitz M (2010) Rethinking digital design: why design must change. IEEE Micro 30(6): 9–24
36. Kim G, Lee MM, Kim J, Lee JW, Abts D, Marty M (2014) Low-overhead network-on-chip support for location-oblivious task placement. IEEE Trans Comput 99:1 PrePrints
37. Itoh K (2009) Adaptive circuits for the 0.5-V nanoscale CMOS Era. In: ISSCC
38. International Technology Roadmap for Semiconductors 2011. System Drivers, Figure SYSD3
39. Howard J et al (2010) A 48-core IA-32 message-passing processor with DVFS in 45nm CMOS. In: ISSCC, pp 108–109
40. Kim W, Gupta MS, Wei GY, Brooks D (2008) System level analysis of fast, per-core DVFS using on-chip switching regulators. In: International symposiun on high-performance computer, architecture
41. Isci C, Buyuktosunoglu A, Cher C, Bose P and Martonosi M (2006) An analysis of efficient multi-core global power management policies: maximizing performance for a given power budget. In: International symposium on microarchitecture, pp 347–358
42. Jain R, Geuskens B, Kim S, Khellah M, Kulkarni J, Tschanz J, De V (2014) A 0.45-1V fully-integrated distributed switched capacitor DC-DC converter with high density MIM capacitor in 22 nm tri-gate CMOS. IEEE J Solid-State Circuit PP(99):1–11
43. Robert Hilbrich J, van Kampenhout R (2011) Partitioning and task transfer on noC-based many-core processors in the avionics domain. Softwaretechnik-Trends 31(3)
44. Trivio Francisco, Snchez Jos L, Alfaro Francisco J, Flich Jos (2012) Network-on-chip virtualization in chip-multiprocessor systems. J Syst Archit Embed Syst Design 58(3–4):126–139
45. Sem-Jacobsen FO, Rodrigo Mocholi S, Strano A, Skeie T, Bertozzi D and Gilabert F (2013) Enabling Power Efficiency through Dynamic Rerouting On-Chip. ACM Trans Embed Comput Syst (TECS) 12(4):1–111:23
46. Strano A, Ludovici D, Pavlidis V, Angiolini F, Krstic M, Bertozzi D (2011) Synchronization Challenge, chapter 6. In: Flich J, Bertozzi D (eds) Designing network on-chip architectures in the nanoscale era. Chapman and Hall/CRC Press, London Taylor and Francis [distributor]
47. Loi I, Angiolini F, Benini L (2008) Developing mesochronous synchronizers to enable 3D NoCs. In: Proceedings of the design, automation and test in Europe conference, pp 1414–1419
48. Saponara S, Cecchini T, Sechi F, Fanucci L (2009) Pin-limited frequency converter IP bridge for efficient communication of automotive IC sensors with off-chip ECUs. In: IEEE international workshop on intelligent data acquisition and advanced computing systems: technology and applications, pp 167–171
49. Tatenguem HF et al (2011) Contrasting multi-synchronous MPSoC design styles for fine-grained clock domain partitioning: the full-HD video playback case study. In: Proceedings of the 4th international workshop on network on chip architectures, pp 37–42

50. Krstic M et al (2012) Evaluation of GALS methods in scaled CMOS technology: moonrake chip experience. IJERTCS 3(4):1–18
51. Ludovici D, Strano A, Bertozzi D, (2009) Architecture design principles for the integration of synchronization interfaces into network-on-chip switches. In: 2nd international workshop on network on chip architectures, pp 31–36
52. Vangal S et al (2007) An 80-tile 1.28TFLOPS network-on-chip in 65nm CMOS. In: ISSCC, pp 98–589
53. Vivek De et al (2014) A 340mV-to-0.9V 20.2Tb/s source-synchronous hybrid packet/circuit-switched 1616 network-on-chip in 22nm Tri—Gate CMOS. In: ISSCC
54. Nowick SM, Singh M (2011) High-performance asynchronous pipelines: an overview. IEEE Design Test Comput 28(5):8–22
55. Yakovlev A, Vivet P, Renaudin M (2013) Advances in asynchronous logic: from principles to GALS & NoC, recent industry applications, and commercial CAD tools. In: Design, automation & test in Europe conference & exhibition (DATE), pp 1715–1724
56. Moreira MT, Magalhaes FG, Gibiluka M, Hessel FP, Calazans NLV (2013) BaBaNoC: an asynchronous network-on-chip described in Balsa. In: International symposium on rapid system prototyping (RSP), pp 37–43
57. Lee W, Vij VS, Thatcher AR, Stevens KS Design of low energy, high performance synchronous and asynchronous 64-point FFT. In: DATE '13 proceedings of the conference on design, automation and test in, Europe, pp 242–247
58. Plana LA et al (2011) SpiNNaker: design and implementation of a GALS multicore system-on-chip. ACM JETC 7(4):17:1–17:18
59. Thonnart Y, Vivet P, Clermidy F (2010) A fully-asynchronous low-power framework for GALS NoC integration. In: DATE, pp 33–38
60. Beerel PA, Dimou GD, Lines AM (2011) Proteus: an ASIC flow for GHz asynchronous designs. IEEE Design Test Comput 28(1):36–51
61. Thonnart Y, Beigne E, Vivet P (2012) A pseudo-synchronous implementation flow for WCHB QDI asynchronous circuits. In: 18th IEEE international symposium on asynchronous circuits and systems (ASYNC), pp 73–80
62. Gebhardt D, You J, Stevens KS (2011) Design of an energy-efficient asynchronous NoC and its optimization tools for heterogeneous SoCs. IEEE Trans Comput Aided Design Integr Circuits Syst 30(9):1387–1399
63. Imai M, Yoneda T, (2011) Improving dependability and performance of fully asynchronous on-chip networks. In: 17th IEEE international symposium on asynchronous circuits and systems (ASYNC), pp 65–76
64. Ghiribaldi A, Bertozzi D, Nowick SM (2013) A transition-signaling bundled data NoC switch architecture for cost-effective GALS multicore systems. In: Design, automation & test in Europe conference & exhibition (DATE), pp 332–337
65. Clermidy F et al (2010) MAGALI: a network-on-chip based multi-core system-on-chip for MIMO 4G SDR. In: IEEE international conference on IC design and technology (ICICDT), pp 74–77
66. Beigne E et al (2009) An asynchronous power aware and adaptive NoC based circuit. IEEE J Solid-State Circuits 44(4):1167–1177
67. ETP4HPC Strategic Research Agenda. http://www.etp4hpc.eu/
68. International technology roadmap for semiconductors 2011. Interconnect
69. Pasricha S, Dutt N (2008) Trends in emerging on-chip interconnect technologies. IPSJ Trans Syst LSI Design Methodol 1:2–7
70. Carloni LP et al (2009) Networks-on-chip in emerging interconnect paradigms: advantages and challenges. In: Proceedings of 3rd ACM/IEEE international symposium networks-on-chip, pp 93–102
71. Kachris Christoforos, Tomkos Ioannis (2012) A survey on optical interconnects for data centers. IEEE Commun Surv Tutor 14(4):1021–1036
72. Kirman N et al (2006) Leveraging optical technology in future bus-based chip multiprocessors. In: MICRO
73. Batten C et al (2008) Building manycore processor-to-dram networks with monolithic silicon photonics. In: Hot interconnects, pp 21–30
74. Pan Y, Kumar P, Kim J, Memik G, Zhang Y, Choudhary AN (2009) Firefly: illuminating future network-on-chip with nanophotonics. In: ISCA, pp 429–440
75. Vantrease D, Binkert N. L, Schreiber R, Lipasti MH (2009) Light speed arbitration and flow control for nanophotonic interconnects. In: MICRO'09, pp 304–315
76. Kurian G, Miller JE, Psota J, Eastep J, Liu J, Michel J, Kimerling LC, Agarwal A (2010) ATAC: a 1000-core cache-coherent processor with on-chip optical network. In: PACT'10, pp 477–488
77. Vantrease D et al (2008) Corona: system implications of emerging nanophotonic technology. In: ISCA

78. Cianchetti MJ, Kerekes JC, Albonesi DH (June 2009) Phastlane: a rapid transit optical routing network. SIGARCH Comput. Archit. News 37:441–450
79. Pan Y, Kim J, Memik G (2010) Flexishare: channel sharing for an energy-efficient nanophotonic crossbar. In: HPCA, pp 1–12
80. Shacham A, Lee BG, Biberman A, Bergman K, Carloni LP (2007) Photonic NoC for DMA communications in chip multiprocessors. In: Hot interconnects
81. Chan J, Hendry G, Biberman A, Bergman K, Carloni LP (2010) Phoenixsim: a simulator for physical-layer analysis of chip-scale photonic interconnection networks. DATE
82. Hendry G et al (2009) Analysis of photonic networks for a chip multi-processor using scientific applications, DATE, 2010. In: Proceedings of the third international symposium on networks-on-chip (NOCS)
83. Ortn Obn M, Ramini L, Tatanguem Fankem H, Vinals-Yufera V, Bertozzi D (2014) A complete electronic network interface architecture for global contention-free communication over emerging optical networks-on-chip. In: Proceedings of GLSVLSI symposium
84. Ramini L, Grani P, Tatenguem Fankem H, Ghiribaldi A, Bartolini S, Bertozzi D (2014) Assessing the energy break-even point between an optical NoC architecture and an aggressive electronic baseline. In: Proceedings of DATE
85. Kurian G et al (2012) Cross-layer energy and performance evaluation of a nanophotonic manycore processor system using real application workloads. In: IEEE 26th international parallel & distributed processing symposium (IPDPS), pp 1117–1130
86. Chang MF et al (2008) CMP network-on-chip overlaid with multi-band RF-interconnect. In: Proceedings of IEEE international symposium on high-performance computer architecture (HPCA), Feb 16–20, pp 191–202
87. Zhao D, Wang Y (2008) SD-MAC: design and synthesis of A hardware-efficient collision-free QoS-aware MAC protocol for wireless network-on-chip. IEEE Trans Comput 57(9):1230–1245
88. Lee SB et al (2009) A scalable micro wireless interconnect structure for CMPs. In: Proceedings of ACM annual international conference on mobile computing and networking (MobiCom), pp 20–25
89. Kempa K et al (2007) Carbon nanotubes as optical antennae. Adv Mater 19:421–426
90. Deb S, Ganguly A, Pande PP, Belzer B, Heo D (2012) Wireless NoC as interconnection backbone for multicore chips: promises and challenges. IEEE J Emerg Selected Topics Circuits Syst 2(2):228–239
91. Weiss SM, Molinari M, Fauchet PM (2003) Temperature stability for silicon-based photonic band-gap structures. Appl Phys Lett 83(10):1980–1982
92. Yu X et al (2011) A wideband body-enabled millimeter-wave transceiver for wireless network-on-chip. In: Proceedings of the 54th IEEE midwest symposium on circuits and system, pp 1–4
93. Lee SB et al (2009) A scalable micro wireless interconnect structure for CMPs. In: Proceedings of the ACM annual international conference on mobile computing and network. (MobiCom), pp 20–25
94. Ganguly A et al (2011) Scalable hybrid wireless network-on-chip architectures for multi-core systems. IEEE Trans. Comput 60(10):1485–1502
95. Batten C, Joshi A, Stojanovic V, Asanovic K (2012) Designing chip-level nanophotonic interconnection networks. IEEE J Emerg Selected Topics Circuits Syst 2(2):137–153
96. Leu JC, Stojanovic V, (2011) Injection-locked clock receiver for monolithic optical link in 45nm. Asian solid-state circuits conference, Jeju, Korea, pp 149–152
97. Ramini L, Grani P, Bartolini S, Bertozzi D (2013) Contrasting wavelength-routed optical NoC topologies for power-efficient 3D-stacked multicore processors using physical-layer analysis. In: DATE, pp 1589–1594
98. Le Beux S, Trajkovic J, O'Connor I, Nicolescu G (2011) Layout guidelines for 3D architectures including optical ring network-on-chip (ORNoC). VLSI-SoC, pp 242–247
99. DiTomaso D et al (2011) iWise: inter-routerwireless scalable express channels for network-on-chips (NoCs) architecture. In: Proceedings of the annual symposium on high performance interconnects, pp 11–18
100. Deb S et al (2010) Enhancing performance of network-on-Chip architectures with millimeter-wave wireless interconnects. In: Proceedings of the IEEE international conference on ASAP, pp 73–80
101. Zhao D et al (2011) Design of multi-channel wireless NoC to improve on-chip communication capacity. In: Proceedings of the 5th ACM/IEEE international symposium on networks-on-chip, pp 177–184
102. Wang C et al (2011) A wireless network-on-chip design for multicore platforms. In: Proceedings of the 19th international euromicro conference on parallel, distributed network-based process., pp 409–416
103. Chang K et al (2012) Performance evaluation and design trade-offs for wireless network-on-chip architectures. ACM J Emerg Technol Comput Syst 8:1–23:25
104. Deb S et al (2012) Design of an efficient NoC architecture using millimeter-wave wireless links. In: Proceedings of the IEEE international symposium on quality electron. Design (ISQED), pp 165–172

105. Ganguly A et al (2011) Complex network inspired fault-tolerant NoC architectures with wireless links. In: Proceedings of the 5th ACM/IEEE international symposium on networks-on-chip, pp 1485–1502
106. Ganguly A et al (2011) A unified error control coding scheme to enhance the reliability of a hybrid wireless Network-on-Chip. In: Proceedings of the IEEE international symposium defect fault tolerance VLSI nanotechnology system, pp 277–285
107. Boos A, Ramini L, Schlichtmann U, Bertozzi D (2013) PROTON: an automatic place-and-route tool for optical Networks-on-Chip. ICCAD, pp 138–145
108. Deb S, Chang K, Ganguly A, Pande P (2010) Comparative performance evaluation of wireless and optical NoC architectures. SoCC 487–492
109. Krzanich B CES 2014 Keynote. http://www.intel.com/content/www/us/en/events/intel-ces-keynote.html