Cost-Effective and Flexible Asynchronous Interconnect Technology for GALS Systems

Davide Bertozzi, Gabriele Miorandi, and Alberto Ghiribaldi, *University of Ferrara, Ferrara, 44122, Italy*
Wayne Burleson, *University of Massachusetts Amherst, Amherst, MA, 01003, USA*
Greg Sadowski, *Advanced Micro Devices, Inc., Boxborough, MA, 01719, USA*
Kshitij Bhardwaj, Weiwei Jiang, and Steven M. Nowick, *Columbia University, New York, NY, 10027, USA*

In this article, a novel interconnect technology is presented for the cost-effective and flexible design of asynchronous networks-on-chip. It delivers asynchrony in heterogeneous system integration while yielding low-energy on-chip data movement. The approach consists of both a lightweight asynchronous switch architecture (using transition-signaling protocols and bundled-data encoding) and a complete synthesis flow built on top of mainstream industrial CAD tools. For the first time, this article demonstrates compelling area, performance and power benefits when compared to a recent commercial synchronous switch, and the ability of the tool flow to correctly instantiate a complete and competitive network topology.

Current computing architectures bear less and less resemblance to the early multicore processors. On the one hand, critical design challenges are being tackled, ranging from the utilization wall and dark silicon issues[1] to power/thermal management and reliability.[2] On the other hand, fundamental shifts from traditional von Neumann architectures are gaining traction.[3] Among them, spiking neural-network-based neuromorphic systems[4,5] use biological inspiration and obtain energy efficiency by exploiting asynchronous event-driven computation.

From the system design viewpoint, the common challenge to the above trends consists of integrating a large number of fine-grain computational units while decoupling their operating mechanisms and conditions.

This challenge motivates the recent surge of interest, in industry and academia, in globally-asynchronous locally-synchronous (GALS) architectures, and the design of asynchronous networks-on-chip (NoCs) to support them.[6] In a GALS system, cores are local islands of synchronicity that interact over a fully asynchronous interconnection network.

Compared to synchronous counterparts, asynchronous NoCs bring several potential advantages:

- No overhead of global clock distribution, tuning, and management.
- No need for performance equalization within individual unbalanced pipelines, and across different pipelines, in the network, leading to aggregate system-level performance benefits.
- Support for optimized flit-level performance, tailored to the different timing paths that each flit-type activates, unlike traditional worst-case clocked design.

However, despite their promise, two main barriers still prevent asynchronous NoCs from fulfilling modern optimization, scaling and flexibility requirements, thus limiting applicability.

First, the choice of communication protocols and data-encoding schemes in most state-of-the-art asynchronous NoCs aims to simplify hardware design (e.g., using four-phase, or "return-to-zero," protocols) and to enforce extreme timing robustness (e.g., using "delay-insensitive" data encoding), at the cost of low throughput, high area occupancy, poor coding efficiency, and high energy-per-bit.

0272-1732 © 2020 IEEE
Digital Object Identifier 10.1109/MM.2020.3002790
Date of publication 16 June 2020; date of current version 20 January 2021.

ASYNCHRONOUS COMMUNICATION CHANNELS: PROTOCOLS AND DATA ENCODING

Asynchronous components communicate via clockless handshaking, which involves defining both a *handshaking protocol* and a *data-encoding scheme.*[1,2]

There are two common handshaking protocols. *Four-phase handshaking* ("return-to-zero") requires two round-trip communications per transaction [see Figure S1 (a)], but potentially leads to simpler hardware, since signals return to a baseline value (i.e., 0) between transactions. In contrast, in *two-phase handshaking* ("non-return-to-zero," or *transition- signaling*), each control signal makes a single toggle, with no return-to-zero phase, incurring only one round-trip communication per transaction [see Figure S1(b)]. Hence, two-phase protocols are preferred for high-performance circuits, though they may lead to more complex hardware. A key challenge addressed by the current research is to employ two-phase handshaking extensively in the NoC switch while retaining low hardware overhead.

The most common data-encoding schemes are *delay-insensitive (DI) codes* and *single-rail bundled data. DI codes* support robust communication by explicitly encoding both data validity and actual data values. Most common is *dual-rail encoding* [see Figure S1(c)], where each bit is encoded with two rails or wires. Independent of transmission time or relative bit skew, the receiver can unambiguously identify when each bit is valid using a completion detector. Overall, these codes provide great resilience to physical and operating variability. However, most DI schemes have poor coding efficiency and high energy-per-bit, due to their wiring overhead.

Alternative data-encoding schemes, such as *single-rail bundled-data* [see Figure S1(d)], use moderate timing constraints, while offering high coding efficiency and low energy-per-bit. This approach uses a standard synchronous-style single-rail data channel with binary data encoding. An extra request ("*req*") wire is then "bundled" with the data, serving as a local strobe on demand, whenever data are sent, along with a backwards acknowledgment ("*ack*") wire. This scheme has the benefit of allowing the use of synchronous-style, i.e., hazardous, computation blocks. Both four-phase and two-phase protocols are common.[1,2] For correct implementation, a single one-sided relative timing constraint (RTC) must be satisfied, that the *req* delay is always longer than worst case data transmission. This *bundling constraint* is typically met by inserting a small matched delay on the control line, when needed.[2] Unlike synchronous timing, however, such constraints are localized: there is no global timing constraint, and unbalanced stages can correctly interact with their own matched delays. However, current commercial CAD tools offer poor support for these RTCs, since they target min/max delay constraints with absolute timing only.

REFERENCES

1. S. M. Nowick and M. Singh, "Asynchronous design - part 1: Overview and recent advances," *IEEE Des. Test*, vol. 32, no. 3, pp. 5–18, Jun. 2015.
2. S. M. Nowick and M. Singh, "High-performance asynchronous pipelines: An overview," *IEEE Des. Test Comput.*, vol. 28, no. 5, pp. 8–22, Sep./Oct. 2011.

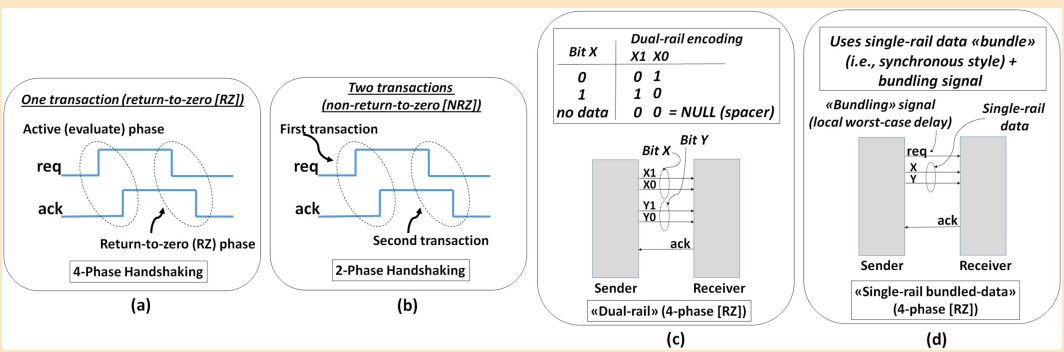

FIGURE S1. Asynchronous (a-b) handshaking protocols, and (c-d) data-encoding schemes.

ASYNCHRONOUS NOC ARCHITECTURES AND SYNTHESIS TOOL FLOWS

Most asynchronous NoC architectures target highly robust design techniques to facilitate timing closure, composability, and tolerance of physical and operational delay variations. These designs use delay-insensitive (DI) codes on data channels, and a so-called "quasi-delay-insensitive" (QDI) style for switch design—whose only timing assumption is that wire forks are "isochronic," i.e., have roughly equal branches.[1,2] While this approach provides ease-of-design, it typically comes at a significant cost in area and power, due to the use of two wires per bit and return-to-zero protocols. As an example, the first generation of a mainstream QDI switch with DI channels, ANoC, reports a 25% energy-per-flit overhead and 80% greater area compared to a synchronous counterpart.[3] However, it still achieves significant savings in total network power (85%) on low-traffic telecom benchmarks, due to its inherent asynchronous ability to exploit sparse activity.

Alternatively, several single-rail bundled-data asynchronous NoCs have been proposed, which incorporate relative timing constraints (RTCs), and show promise in overall cost metrics: coding efficiency, area, power, and performance.[4,5,6,7] However, the development of automated CAD flows for these NoCs is especially challenging. Commercial CAD tools typically support only absolute min/max delay constraints. In contrast, asynchronous RTCs define the required ordering of pairs of control and/or datapath delays (e.g., a control event must occur only after associated data is valid), whose absolute values need not be defined and may depend on later synthesis steps (gate mapping, physical design). A basic iterative synthesis procedure has been proposed,[8] using synchronous CAD tools, but its applicability is currently limited to small subsystems, and no course of action is taken to ensure convergence or to optimize quality of results in the general case.

Finally, while bundled-data NoCs have demonstrated benefits over synchronous NoCs in some cost metrics, their limited optimization currently results in other substantial overheads, especially in area[6] and performance.[6,7] In addition, most bundled-data NoC research rarely goes beyond switch-level analysis. There are a few promising recent exceptions,[7] but the early stage of the hierarchical tool flow for network synthesis prevents the bundled-data NoC from keeping up with performance expectations. The goal of our research is to overcome these overheads, in both switch design and tool development.

REFERENCES

1. J. Bainbridge and S. Furber, "Chain: A delay-insensitive chip area interconnect," *IEEE Micro*, vol. 22, no. 5, pp. 16–23, Sep./ Oct. 2002.
2. A. Lines, "Asynchronous interconnect for synchronous SoC design," *IEEE Micro*, vol. 24, no. 1, pp. 32–41, Jan./ Feb. 2004.
3. Y. Thonnart, P. Vivet, and F. Clermidy, "A fully-asynchronous low-power framework for GALS NoC integration," in *Proc. ACM/IEEE Des., Autom. Test Eur. Conf.*, 2010, pp. 33–38.
4. T. Bjerregaard and J. Sparsoe, "A router architecture for connection-oriented service guarantees in the MANGO clockless network-on-chip," in *Proc. ACM/IEEE Des., Autom. Test Eur. Conf.*, 2005, pp. 1226–1231.
5. R. Dobkin, R. Ginosar, and A. Kolodny, "QNoC asynchronous router," Integr. VLSI J., vol. 42, no. 2, pp. 103–115, 2009.
6. M. Gibiluka, M. T. Moreira, F. G. Moraes, and N. L. V. Calazans, "BAT-Hermes: A transition-signaling bundled-data NoC router," in *Proc. IEEE Latin Amer. Symp. Circuits Syst.*, 2015, pp. 1–4.
7. M. Imai, T. V. Chu, K. Kise, and T. Yoneda, "The synchronous vs. asynchronous NoC routers: An apple-to-apple comparison between synchronous and transition signaling asynchronous designs," in *Proc. ACM/IEEE Int. Symp. Netw.-on-Chip*, 2016, pp. 1–8.
8. M. Gibiluka, M. T. Moreira, and N. L. V. Calazans, "A bundled-data asynchronous circuit synthesis flow using a commercial EDA framework," in *Proc. Euromicro Conf. Digit. Syst. Des.*, 2015, pp. 79–86.

Second, asynchronous NoCs currently suffer from limited computer-aided design (CAD) tool support, due to a disconnect in timing models between clocked and asynchronous designs.

Main Contributions

This article aims at an inflection point in asynchronous NoC design, which relies on two pillars.

First, it presents a new asynchronous switch architecture combining the high performance of *two-phase,* or *"transition-signaling," communication protocols* (i.e., with only one round-trip handshake per transaction) with the coding efficiency of *single-rail bundled-data encoding*. In practice, the datapath consists of synchronous-style "bundles" of single wires per data bit, along with associated req/ack handshaking signals that toggle only once per data transfer, thereby enabling higher throughput (see the "Asynchronous Communication Channels: Protocols and Data Encoding" sidebar).

Second, we developed an automated synthesis and place-and-route flow for the bottom-up hierarchical implementation of bundled-data asynchronous NoCs, leveraging mainstream industrial synchronous CAD tools.

This combination of transition-signaling asynchronous communication and single-rail bundled data has only rarely been used for on-chip interconnection networks. In fact, the fundamental challenges are 1) to master the potential area and complexity overhead of two-phase asynchronous pipelines within the switch, and 2) efficiently to enforce one-sided relative timing constraints (RTCs) on all bundled datapaths (i.e., *bundling constraints*), using automated commercial CAD tools and without overloading the synthesis engine (see the "Asynchronous NoC Architectures and Synthesis Tool Flows" sidebar). In particular, for correct operation, the data wires on each channel must always be valid and stable before the corresponding request is observed at the receiver side.

Using the proposed architecture and tool flow to synthesize a complete 4×4 2-D mesh topology, an asynchronous two-phase bundled-data NoC for the first time is shown to dominate a clock-gated synchronous counterpart for ultra-low power systems[7] under most operating conditions. When projected to the bandwidth requirements of a full HD video playback application, the asynchronous NoC exhibits latency savings up to 37% and total power savings up to 45%.

Finally, in a direct comparison with a recent commercial AMD synchronous router, using identical 14-nm FinFET technology, results confirm substantial benefits: 55% lower area, 28% lower latency, and reductions of 88% idle and 58% active power.

ASYNCHRONOUS BUNDLED-DATA NOC SWITCH ARCHITECTURE

Figure 1(a) shows the top-level view of our asynchronous switch architecture, instantiated with 5 I/O ports for a 2-D mesh topology. The switch is modular and can be scaled to connect an arbitrary number of input port modules (IPMs) and output port modules (OPMs). The figure shows an expanded view of an IPM and an OPM. The interconnecting crossbar is enclosed in the OPM schematic. The architecture implements wormhole routing; hence, packets are processed at the level of individual flow control units (flits).

Mousetrap Asynchronous Pipelines

The switch builds extensively on a high-performance asynchronous pipeline, *Mousetrap*,[8] developed at Columbia University [see structure in Figure 1(b), and detailed comparisons and evaluation in the related paper[8]]. It uses a two-phase protocol and single-rail bundled-data encoding, where the latter provides nearly identical coding efficiency as a synchronous datapath and is fully compliant with a standard-cell design methodology. Each data item advances through the pipeline "elastically," based on local conditions, coordinated by a so-called "capture-pass" handshaking protocol: single-latch registers are *normally transparent,* only closing to protect data immediately after it enters a stage. Once data enters the next stage, the current register is reopened. Mousetrap uses simple control circuits (a single exclusive-NOR gate) and data registers (a single bank of level-sensitive D-latches), with low area and delay overheads.

Asynchronous Switch Design

As shown in Figure 1(a), the switch's buffering includes:

- a single Mousetrap input stage, decoupling the cycle time of the upstream link from the switch;
- an asynchronous circular FIFO on each output port;
- a single Mousetrap internal stage, decoupling the cycle time of the switch from that of the circular FIFO;
- an optional output Mousetrap stage, decoupling the cycle time of the circular FIFO from the downstream link.

Initially, arriving flits of a packet are stored sequentially in the input Mousetrap stage and transmitted through the IPM. The request bundling signal and associated head flit are speculatively broadcast through the crossbar to every OPM.[9] Concurrently, the *Routing Logic* computes the actual target output port for the packet, based on its head flit, which is then stored into a *Memory Element*. The latter asserts a single *Path Allocation Request* to the selected OPM

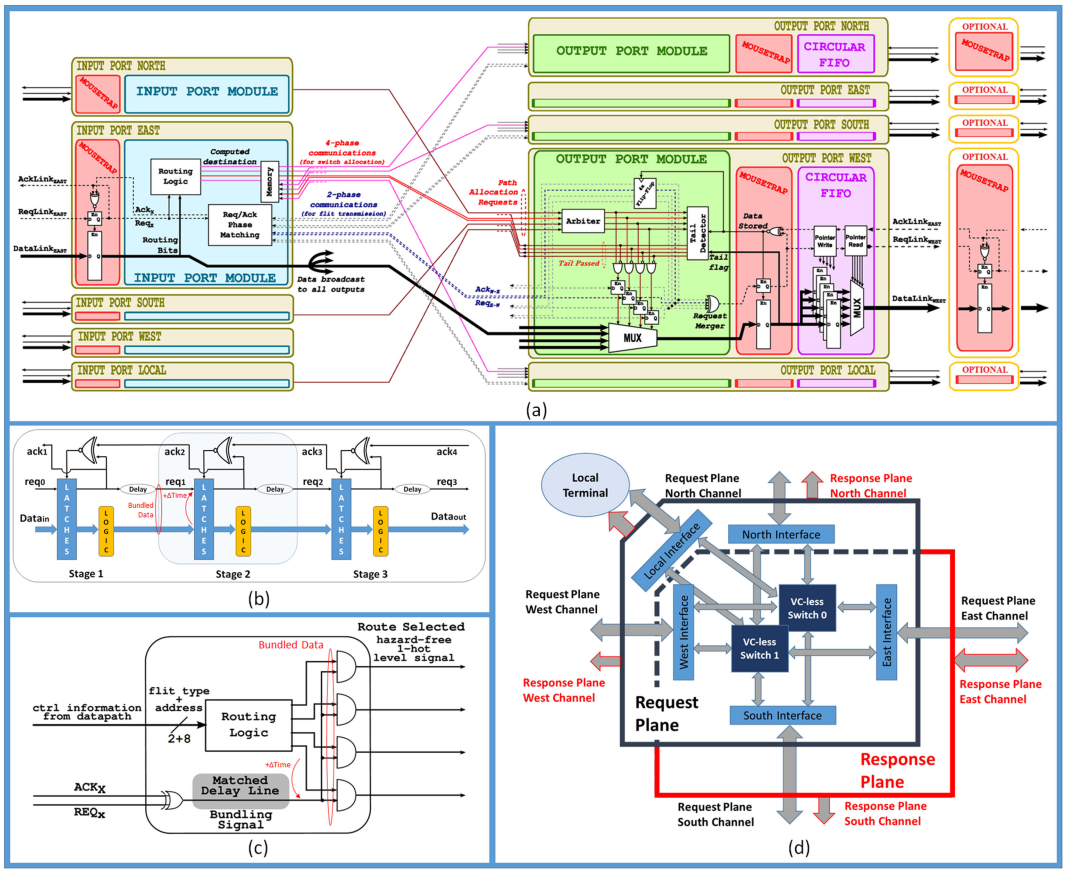

FIGURE. 1. Proposed asynchronous switch architecture. (a) Schematic view. (b) Mousetrap asynchronous pipeline. (c) Routing logic. (d) Switch instance (2 planes, 2 VCs/plane).

for the entire packet transmission time, while other speculative requests are ignored. Once the *allocation request* acquires the target OPM arbiter, the reserved path from input to output channel for the remainder of the packet is normally transparent and free-flowing, unlike most synchronous designs, with latch registers closing only transiently after each flit arrives for flow control.

A key OPM component is the *four-way arbiter*, mediating between competing input requests in continuous time, without any reference clock. This component, the only non-standard cell in the switch, incorporates three small analog mutexes that guarantee a correct resolution,[9] though digital variants have been proposed with slightly degraded mean-time-between-failure (MTBF).[10] The arbiter has been generalized into a scalable family of N-way asynchronous tree arbiters, enabling the design of fully parameterizable N×M switches for generic topologies.[11]

Almost all communication in the design (e.g., intra-switch and inter-switch) uses two-phase signaling. Hence, there is only a single roundtrip crossbar and link communication per flit, which enables higher throughput than the more common four-phase protocol.

The output *circular FIFO* is latency-optimized and comes with low area footprint. It uses a novel parallel microarchitecture, with Mousetrap-based single-latch architectural registers.[9]

Correct switch operation depends on several relative timing constraints on datapath and control.[9] While most constraints are typically satisfied for normal operating conditions, two critical constraints are likely to require additional synthesis effort: bundling constraints between consecutive Mousetrap stages [see Figure 1(b)], especially on links,[8] and one on the routing logic [see Figure 1(c)], where a matched delay line ensures hazard-free operation.[9] Margins enforced on all relative timing constraints are also critical to determining switch latency and throughput.

The proposed architecture targets realistic switch instances, such as the one in Figure 1(d), through the support of physical planes (e.g., dedicated to different

message types) and of virtual channels (VCs) in each plane (e.g., to relieve head-of-line blocking, or for quality-of-service).

For efficient implementation of VCs, control logic overhead is minimized by using a simple replicated copy of the crossbar for each VC, then multiplexing their outputs onto a single physical stream/link via a small flit-level asynchronous arbiter in each I/O interface [see Figure 1(d)].[12] Buffer availability downstream is identified on a VC-basis using a new asynchronous credit-based scheme that has been optimized for throughput.[13] In practice, this "lazy credit update" policy improves performance, deferring unnecessary non-critical credit increment updates, which are queued and take place only with the next credit decrement request.*

Finally, two useful additional capabilities have been developed: 1) a comprehensive built-in self-testing framework,[14] and 2) an FPGA-based switch design and CAD framework, targeting the Xilinx Vivado tool set.[10]

IMPLEMENTATION TOOL FLOW

An automated tool flow for bundled-data NoC implementation using commercial synchronous CAD tools has also been developed. It targets synchronous-equivalent design flexibility, as well as the bottom-up hierarchical synthesis of complete asynchronous NoCs with arbitrary topologies.[15]

Figure 2 provides an overview of the complete tool flow. Synthesis steps are in blue boxes, while place-&-route (P&R) steps are in red boxes. It is structured into a first-stage flow for switch macros (steps 1–10), and a second-stage flow for top-level design of the network as a whole (steps 11–16). Without lack of generality, the Synopsys Design Compiler is used for logic synthesis and the IC Compiler is used for P&R.

The flow revolves around a detailed optimization methodology of *relative timing margins (RTMs)*, structured as a *nested loop*.

First, all datapath delays are locally and individually optimized (in Steps 3 and 8). Then, an inner loop is used as a baseline iterative procedure that fine-tunes control path delays associated with such locally-optimized datapath delays. In order to handle the scale and optimization challenge of complex switches and network topologies, we optionally provide an outer nested loop that hits RTMs gradually. In particular, the RTM is not immediately set to its final target, but to more relaxed intermediate values (e.g., no margin, then 5%, 7%, and finally 10% of the datapath delay to be matched), which the synthesis and the P&R tool can more easily fulfill.

Finally, especially during top-level topology (i.e., link) synthesis, the concurrent convergence of many control signals on the intermediate or target RTM can be optionally further accelerated by an engineering change order (ECO), which selectively places small delay elements on violating control paths.

Unlike prior approaches, this procedure not only guarantees functional correctness (i.e., all RTCs are met) but also gains tight control over RTMs, i.e., it generally prevents the delays on control lines from greatly exceeding the datapath delays to be matched, thus avoiding significant latency and cycle time degradations. Simultaneously, it makes the convergence in satisfying several bundling constraints, in parallel, computationally affordable and effective, without overloading the synthesis engine.

SWITCH-LEVEL ASSESSMENT

Our bundled-data asynchronous switch, called ***TaBuLA*** (**Tra**nsition-Signaling **Bu**ndled-Data **L**ightweight **A**synchronous router),† is synthesized using the proposed automated tool flow and compared to a leading synchronous NoC switch, *xpipes*.[7] The latter's streamlined architecture, which combines instantiation-time flexibility with silicon efficiency, makes it a representative benchmark for the requirements of the ultra-low power embedded computing domain.‡[16]

Both switches are instantiated with the same VC-less configuration and have homogeneous architectures. The synchronous design is synthesized for maximum performance, using a state-of-the-art clock-gating methodology. It reserves 1 clock cycle for switch traversal and 1 cycle for link traversal.

Post-layout results are reported for an ultra-low power 40-nm industrial technology. Since it does not include asynchronous special cells, standard-cell equivalent implementations are used for *TaBuLA*.

Performance Evaluation

Table 1(a) evaluates basic performance metrics. The leftmost columns show results for switch traversal only, i.e., from input port to arrival at the output buffer

*AMD, Inc., Greg Sadowski and Weiwei Jiang, "Self-Timed Router with Virtual Channel Control," US Patent #10,075,383 (2016).

†The acronym suggests "tabula rasa," meaning "blank slate," denoting a "fresh start," without pre-existing ideas, which is a goal of considering asynchronous interconnect technology.

‡Through the *INoCs* startup, recently acquired by *Arteris Inc.*, the research ideas of the *xpipes* framework have become part of one of the largest commercial NoC ventures.

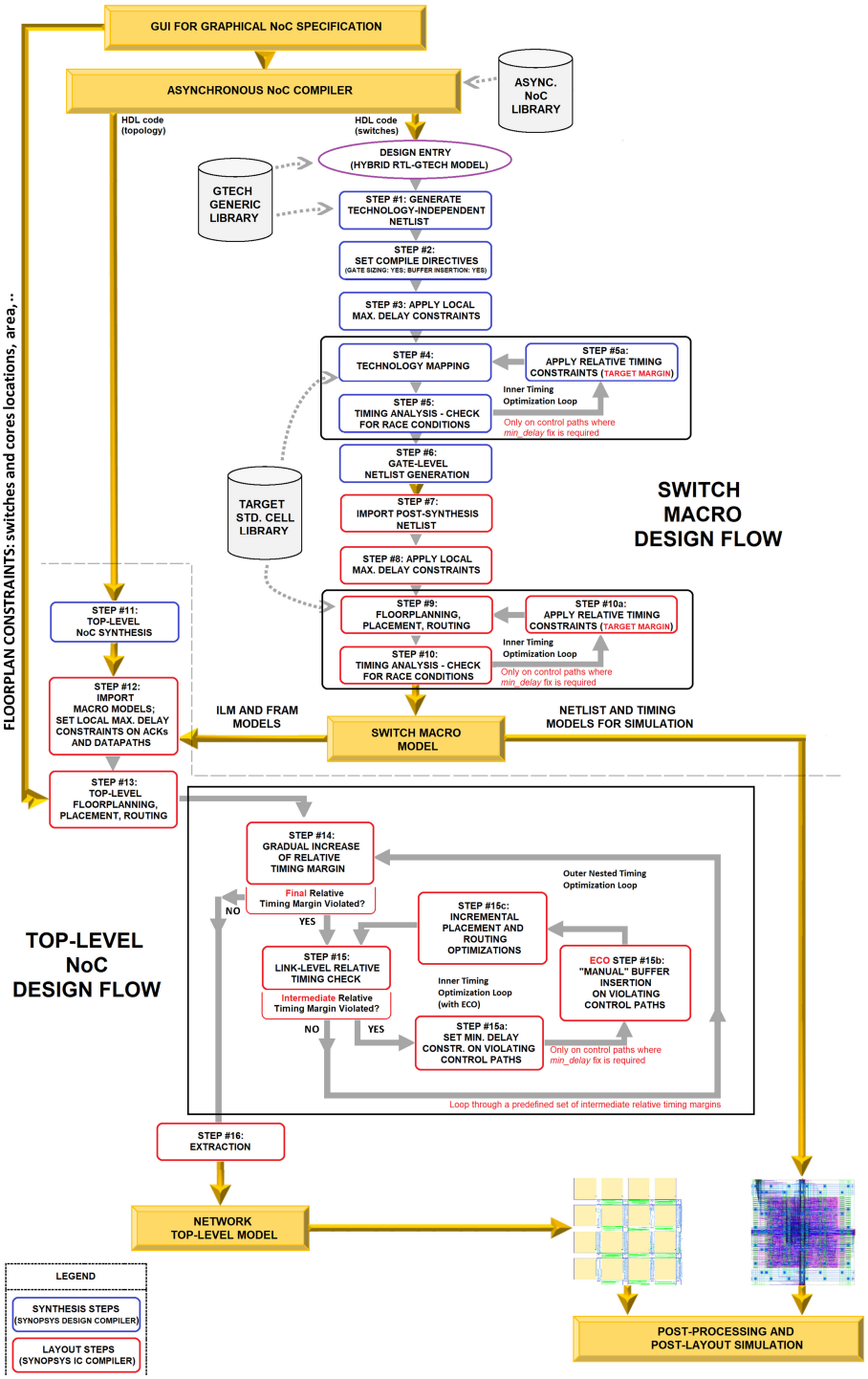

FIGURE 2. Complete bottom-up hierarchical synthesis flow for bundled-data asynchronous NoCs. Steps enclosed in black boxes indicate timing optimization procedures. For simplicity, the figure shows the use of only the inner optimization loop (i.e., iterative convergence directly to the final RTM) for switch synthesis (steps #4-#5-#5a) and P&R (steps #9-#10-#10a), while nested optimization loops (i.e., gradual convergence through intermediate relaxed margins) are used for timing convergence of the topology links (steps #14-#15-#15a-#15b-#15c), combined with ECO steps (#15b) that yield effective and fast timing optimization despite the large number of RTCs to handle at this stage.

TABLE 1. Comparative 5 × 5 switch evaluation. (a) Performance. (b) Area and energy-per-flit.

Performance	Switch Traversal		Synch. Overhead	Switch & Ideal Link Traversal		Synch. Overhead
	Async.	Synch.		Async.	Synch.	
Head Latency (ps)	1165	943	-19%	1579	1886	+19%
Payload Latency (ps)	486	943	+94%	905	1886	+108%
Cycle Time (ps)	934	943	+1%	934	943	+1%

(a)

Quality metrics			Async.	Synch.	Synch. Overhead
Total Switch Area			12433	14266	+15%
Energy-per-flit (pJ)	Continuous injection	3-flit packets	3.88	4.69	+21%
		20-flit packets	3.41	3.72	+9%
	Moderate injection	3-flit packets	3.86	6.51	+69%
		20-flit packets	3.42	5.28	+54%

(b)

inputs. Cycle times are nearly identical, but *TaBuLA*'s payload latency is nearly half of the synchronous latency, since body and tail flits avoid the control logic for routing and switch allocation needed for header flits. However, the synchronous switch exhibits 19% lower head flit latency than *TaBuLA*. This penalty is due to the RTMs, the phase conversion circuit, additional gates for glitch-free operation of the routing logic, and the propagation delay through the decoupling Mousetrap register.

When considering the complete switch and ideal (i.e., zero-delay) link traversals together, the synchronous switch increases latency at the coarser granularity of full clock cycles while *TaBuLA* is more adaptive, and only accounts for the actual link delay (in this case, only through the Circular FIFO). This amplifies the payload latency overhead of *xpipes* (from 94% to 108%) and reverses its comparative head latency (from –19% to +19%). The impact of realistic link delays will be assessed later in the network-level analysis.

Cost Analysis

As listed in Table 1(b), despite the architectural homogeneity, *xpipes* consumes 15% more area than *TaBuLA*.

Table 1(b) also reports total energy-per-flit results (static and dynamic), showing a synchronous overhead ranging from 9% to 21% for continuous injection, and from 54% to 69% for moderate injection. *TaBuLA*'s energy-per-flit is largely insensitive to the traffic scenario, since *TaBuLA* burns energy mainly for productive switching activity and not for idle resources. In contrast, synchronous energy-per-flit increases as the injection rate decreases, since fixed clock-tree power contributions are divided over a lower traffic volume.

COMPARISON WITH STATE OF THE ART

Table 2 compares the performance, area and energy-per-bit of recent asynchronous NoC switches using three state-of-the-art asynchronous design styles:

- ***TaBuLA***, the proposed two-phase bundled-data approach;
- ***ANOC***, a QDI switch using delay-insensitive four-phase communication channels;
- ***BAT-Hermes***, an alternative two-phase bundled-data framework.

The QDI *ANOC*, an influential design series from CEA-LETI, has gone through several generations of technology, implementation, and architecture. A high-quality recent generation[17] is considered in Table 2.

When looking at absolute numbers, a comparable *TaBuLA* switch, also using two physical channels, performs consistently better than *ANOC* in all metrics: 11% lower latency, 20% higher throughput, 82% lower area, and 83% lower energy-per-bit.

After technology and threshold voltage normalization,[§] latencies are roughly comparable, but *TaBuLA* exhibits roughly 10% higher throughput, despite its earlier stage of tool development. *ANOC* largely offsets the performance penalty of its four-phase communication protocol through deep pipelining of its completion detection logic, and using the latest generation of its timing optimization tool flow, but is nonetheless unable to close the gap with the two-phase *TaBuLA*.

Significant advantages, however, appear in the remaining metrics, after normalization, where *TaBuLA* has 72% lower energy-per-bit and 52% lower area. These results clearly correlate to its use of bundled-data encoding and two-phase handshaking protocols.

In contrast to *TaBuLA*, some of *ANOC*'s design-space decisions are oriented to extreme robustness: resilience to arbitrary bit skew, and to high physical

[§]The fanout-of-4 (FO4) delay metric (estimated from technology databooks) is used for performance normalization. The scaling ratio of feature size is applied for area and energy-per-bit, assuming identical supply voltages. When comparing multi-Vth *ANOC* with standard-Vth *TaBuLA* technology, scaled area numbers are slightly pessimistic for *ANOC*, while *ANOC*'s scaled energy-per-bit is optimistic.

TABLE 2. Comparison of asynchronous NoC switches.

5x5 Router with 2 Physical Channels			
Architecture	QDI ANOC in advanced 3D MIMO platform (data from [17])	**TaBuLA 2-phase bundled-data**	BAT-HERMES 2-phase bundled-data (data from [18])
Technology	65nm, 1.2V mixed Vth	**40nm, 1.2V standard Vth**	65nm, 1V standard Vth
Flit width	32 bits	**32 bits**	16 bits
Configuration	source-based routing; 2D router + 3D router#	**distributed routing**	distributed routing
Avg. forward latency (ns)*	1.10 *(2D router only)*	**0.98**	4.54
Equivalent speed (MHz)	890 *(2D router only)*	**1070**	385
Energy/bit (pJ/bit)†	0.69 *(2D router only)*	**0.12**	0.103 (best) 0.135 (worst)§
Area (μm^2)	136220‡ (2D router only)	**24866**	60026

Notes

**Average flit latency of 9-flit packets for ANOC and TaBuLA; packet length unspecified for BAT-Hermes.*
†Measurement methodologies are compatible, except for BAT-HERMES, which uses overly long conflict-free packets, thus underrepresenting switch allocation energy.
‡Extrapolated from the reported number of gates per router.
#Architecture targeted to 3D chips, with both 2D (horizontal) and 3D (vertical) routers; the former have higher performance, and are the focus of this comparison.
§Best: all body flits are zero. Worst: each body flit is inverted with respect to the previous one.

and operational variability, as well as to simplify large-scale physical design. Such an approach can result in over-design for many practical systems. *TaBuLA's* less conservative style leads to major overall cost benefits, as shown above, while still leaving the designer with the flexibility to locally fine-tune timing margins for the requirements of the system at hand.

It is also noteworthy that *ANOC* crossbars, using delay-insensitive codes, are much more wire-intensive, with roughly twice as many wires as in *TaBuLA*. Hence, *TaBuLA* is better suited for environments requiring high-radix switches, such as irregular topologies, 3-D stacking or neuromorphic computing.

Finally, Table 2 presents an unscaled comparison to *BAT-Hermes*,[18] a 16-bit two-phase bundled-data switch. Despite its doubled flit width, *TaBuLA* has 58% lower area, 78% lower latency, and 2.8× equivalent speed (in MFlit/sec), with comparable energy-per-bit.

To normalize results, in addition to technology scaling, *BAT-Hermes* also requires supply voltage and flit width equalization for direct comparison.** Our projections indicate that *TaBuLA*'s savings over *BAT-Hermes* are still as large as 32% for area, 65% for latency, from 18% to 38% for energy-per-bit, with 70% higher throughput. These benefits are attributed to *TaBuLA*'s lightweight control logic and more efficient timing optimization tool flow.

NETWORK-LEVEL ASSESSMENT

For network-level evaluation, one complete *xpipes*-based synchronous 4×4 2-D-mesh NoC (with clock gating) and two asynchronous 4×4 2-D-mesh *TaBuLA* NoCs are laid out in identical 40 nm technology. The

**We optimistically assume that latency and throughput are preserved as flit width is increased from 16 to 32 bits; while for area, we project a 60% overall increment, from similar flit width scaling experiments on *TaBuLA*. Supply voltage is scaled through the popular Alpha-Power Law MOS model. FO4 and dimension-scaling ratios are used to normalize performance and area/energy-per-bit to the same process node, respectively.

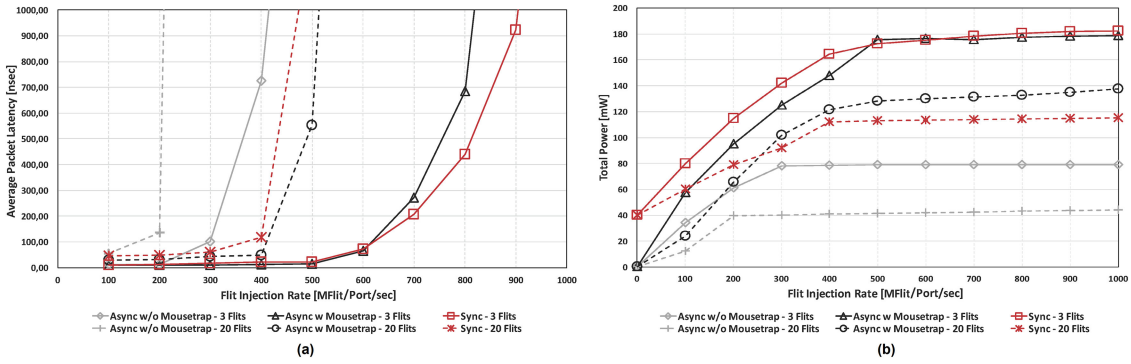

FIGURE 3. Comparative network evaluation. (a) Load curves. (b) Power consumption.

two asynchronous variants are with (*pipelined*) and without (*unpipelined*) an additional output Mousetrap stage decoupling the cycle time of circular FIFOs from the downstream links. The interswitch link length is set to 1 mm, reflecting typical tile sizes in ultra-low power processing platforms. Tiles are modeled as hard non-routable obstructions.[15]

Performance Analysis

The synchronous NoC achieves timing closure at 1 GHz without any guardband. In contrast, the asynchronous NoCs are conservatively synthesized with a 10% RTM; hence, the asynchronous results are more pessimistic.

Load curves are reported in Figure 3(a) for different packet lengths with uniform random traffic. When compared with the synchronous design, *pipelined-TaBuLA* exhibits its full potential when the packet length is long enough, e.g., 20 flits in this study. Here, performance is dominated by the flow-through operation capability of payload flits, where the asynchronous switches show substantial benefits, achieving 36% lower zero-load latency (45.56 ns for synchronous, 28.92 ns for asynchronous) and entering the saturation region at an injection rate that is 33% higher, over the synchronous network.

In contrast, when short (e.g., 3-flit) packets are used, *pipelined-TaBuLA's* performance is mainly determined by head flits. The latency of asynchronous head flits is greater than in the synchronous design, due to the effects of link handshaking, relative timing margins, and propagation through additional Mousetrap stages. Still, the asynchronous NoC provides a 9% lower zero-load latency for delivering an entire packet (10.79 ns for synchronous, 9.83 ns for asynchronous), which improves to 33% lower latency at an injection rate of 500 MFlit/port/s, after which both solutions begin saturating. Finally, the synchronous NoC has a better saturation rate in an extreme region that typically lacks practical interest.

Power Analysis

Total power results are illustrated in Figure 3(b). At low injection rates, the power saving capability of both asynchronous NoCs (roughly 40 mW) is out-of-reach even for the clock-gated synchronous counterpart.

With higher traffic, differentiations in design overhead between *xpipes* and *pipelined-TaBuLA* result in distinct trends for short and long packets.

The power of the synchronous switch is sensitive to its much larger arbitration overhead, clock-tree power and more expensive FSM-based flow control. For the latter, though *xpipes* handles back-pressure through a standard low-overhead stall/go protocol, it requires at least double buffers to avoid dropping incoming packets in case of a stall from the downstream receiver.

In contrast, *TaBuLA* has built-in asynchronous support for flow control through its simple req/ack signaling protocol, hence a single Mousetrap buffer is sufficient for correct operation. *TaBuLA's* power overhead is mainly dominated by the four Mousetrap registers that each flit must traverse per switch, as opposed to two registers in *xpipes*, while its lightweight arbitration contributes only negligible overhead.

On balance, with 3-flit packets, *pipelined-TaBuLA* shows consistent improvement over *xpipes*, with 28%, 17%, 12% and 10% lower total power, respectively, at injection rates of 100, 200, 300 and 400 MFlit/port/s, before the onset of saturation effects.

With 20-flit packets, however, the overall contribution of arbitration is smaller, leading to a more gradual synchronous curve. Here, *pipelined-TaBuLA* outperforms *xpipes* up to a power break-even point at 65% of the maximum injection rate, beyond which the saturation of the synchronous NoC begins.

Asynchronous total power savings are significant, ranging from 17% to 60% over the synchronous version, as traffic decreases from near break-even (200 MFlit/port/s) to lower injection rates (100 MFlit/port/s). Above the break-even point, *xpipes* saves from 10% to 8% when moving up from 300 MFlit/port/s to the start of its saturation region. Further asynchronous improvements are anticipated, since the current design is not fully optimized for switching activity in the routing logic and circular FIFOs.

Finally, the *unpipelined-TaBuLA NoC* provides a radical performance-power tradeoff. While it enters the saturation region at an injection bandwidth that is roughly one third of that of the synchronous NoC for both packet lengths, it offers substantial power savings, ranging from 57% to 45% for 3-flit packets, and from 80% to 50% for 20-flit packets, when moving from injection rates of 100 MFlit/port/sec to saturation. These results make it attractive for low-end embedded systems.

Realistic Traffic

Projected bandwidth requirements are also evaluated for a full HD video playback application for high-end mobile devices,[19] with 1920×1080 pixels at 60 frames/s. The application has 19 communication flows with average bandwidth ranging from 0.1 to 500 MB/s. Even injecting at the maximum data rate of 125 MFlit/s from each switch's local port (although demanded by only 5 of the 19 flows in the original application), and adding the head flit overhead, the asynchronous NoC is largely dominating. With 3-flit and 20-flit packets, power savings are 18% and 45%, respectively, while latency savings are 22% and 37%.

Finally, *TaBuLA* can handle the communication challenges posed by future high-performance data analytics in Edge computing platforms. In particular, it can sustain the communication bandwidth requirements of a pool of 16 inference engines, such as the NVDLA deep neural network accelerator,[20] with power savings of 18% and 10% over *xpipes*, respectively, as the number of MAC units per engine increases from 32 to 64. In addition, *TaBuLA* obtains latency savings of 22% and 40%, respectively.

VALIDATION IN AN INDUSTRIAL ENVIRONMENT

A prototype of the *pipelined-TaBuLA* switch was implemented at AMD Research and compared directly to a commercial synchronous switch (hereafter named *CommSw*),[13] used to handle system-level configuration and power/performance monitoring and control in recent high-end processor and graphics products.

This evaluation is the first reported direct comparison of asynchronous and commercial synchronous NoC switches in identical advanced (14-nm FinFET) technology.

To match *CommSw*'s microarchitecture, *TaBuLA* was expanded into a 2-plane, 2-VC-per-plane implementation [see Figure 1(d)]. The three-cycle *CommSw* was synthesized for the target speed of 1 GHz.[††]

The asynchronous switch was implemented using synchronous design and validation tools of the industrial partner, but limiting the reinstrumentation effort of the stable existing flow as much as possible (e.g., our gradual convergence option could not be used).[13] Due to the use of advanced commercial technology, only post-synthesis results can be reported, but actual parasitics of standard cells were nonetheless imported for accurate power analysis.

The asynchronous router has 55% lower area, 28% lower latency, and 88% and 58% savings in idle and active power, respectively. Most of these savings are due to the use of single-latch-based Mousetrap registers (with small area footprint, and low energy and critical-path latency), lack of global clock distribution, and on-demand activation.

THIS EVALUATION IS THE FIRST REPORTED DIRECT COMPARISON OF ASYNCHRONOUS AND COMMERCIAL SYNCHRONOUS NOC SWITCHES IN IDENTICAL ADVANCED (14-NM FINFET) TECHNOLOGY.

CONCLUSIONS

Emerging computing architectures call for increasing levels of asynchrony during system-level integration. This article proposes a novel asynchronous interconnect technology, *TaBuLA*, which can fulfill this requirement with cost metrics (area, energy-per-bit) that are largely out-of-reach for mainstream synchronous counterparts, while preserving or improving performance depending on the operating conditions.

[††]*CommSw* also has functionality for error detection and configuration, which contributes only a 1%–4% area and power increase, with negligible performance impact.

ACKNOWLEDGMENTS

This work was supported in part by the "Fondo Giovani" Ph.D. program (Italian Government), in part by an FIR Grant (University of Ferrara), in part by the National Science Foundation (NSF) under Grants CCF-1219013 and CCF-1527796, and in part by AMD under a grant from the DOE Exascale Program. This work was completed while Prof. Burleson was at Advanced Micro Devices, Inc.

REFERENCES

1. H. Esmaeilzadeh, E. Blem, R. St. Amant, K. Sankaralingam, and D. Burger, "Dark silicon and the end of multicore scaling," *IEEE Micro*, vol. 32, no. 3, pp. 122–134, May-Jun. 2012.
2. R. G. Kim *et al.*, "Imitation learning for dynamic VFI control in large-scale manycore systems," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* , vol. 25, no. 9, pp. 2458–2471, Sep. 2017.
3. J. Cong, M. A. Ghodrat, M. Gill, B. Grigorian, K. Gururaj, and G. Reinman, "Accelerator-rich architectures: Opportunities and progresses," in *Proc. ACM/IEEE Des. Autom. Conf.*, 2014, pp. 1–6.
4. F. Akopyan *et al.*, "TrueNorth: Design and tool flow of a 65 mW 1 million neuron programmable neurosynaptic chip," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 34, no. 10, pp. 1537–1557, 2015.
5. M. Davies *et al.*, "Loihi: A neuromorphic manycore processor with on-chip learning," *IEEE Micro*, vol. 38, no. 1, pp. 82–99, Jan.-Feb. 2018.
6. M. Krstic, E. Grass, F. K. Gürkaynak, and P. Vivet, "Globally asynchronous, locally synchronous circuits: Overview and outlook," *IEEE Des. Test Comput.*, vol. 24, no. 5, pp. 430–441, Sep.-Oct. 2007.
7. S. Stergiou, F. Angiolini, S. Carta, L. Raffo, D. Bertozzi, and G. De Micheli, "xpipes lite: A synthesis oriented design library for networks on chips," in *Proc. ACM/IEEE Des. Autom. Test Eur. Conf.*, 2005, pp. 1188–1193.
8. M. Singh and S. M. Nowick, "MOUSETRAP: High-speed transition-signaling asynchronous pipelines," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 15, no. 6, pp. 684–698, Jun. 2007.
9. A. Ghiribaldi, D. Bertozzi, and S. M. Nowick, "A transition-signaling bundled data NoC switch architecture for cost-effective GALS multicore systems," in *Proc. ACM/IEEE Des. Autom. Test Eur. Conf.*, 2013, pp. 332–337.
10. K. Bhardwaj, P. Mantovani, L. P. Carloni, and S. M. Nowick, "Towards a complete methodology for synthesizing bundled-data asynchronous circuits on FPGAs," in *Proc. IEEE Int. Symp. Low-Power Electron. Des.*, 2019, pp. 1–6.
11. G. Miorandi, D. Bertozzi, and S. M. Nowick, "Increasing impartiality and robustness in high-performance N-way asynchronous arbiters," in *Proc. IEEE Int. Symp. Asynchronous Circuits Syst.*, 2015, pp. 108–115.
12. G. Miorandi, A. Ghiribaldi, S. M. Nowick, and D. Bertozzi, "Crossbar replication vs. sharing for virtual channel flow control in asynchronous NoCs: A comparative study," in *Proc. IFIP/IEEE Conf. Very Large Scale Integr.*, 2014, pp. 1–6.
13. W. Jiang, D. Bertozzi, G. Miorandi, S. M. Nowick, W. Burleson, and G. Sadowski, "An asynchronous NoC router in a 14nm FinFET library: Comparison to an industrial synchronous counterpart," in *Proc. ACM/IEEE Des. Autom. Test Eur. Conf.*, 2017, pp. 732–733.
14. G. Miorandi, A. Celin, M. Favalli, and D. Bertozzi, "A built-in self-testing framework for asynchronous bundled-data NoC switches resilient to delay variations," in *Proc. ACM/IEEE Int. Symp. Network-on-Chip*, 2016, pp. 1–8.
15. G. Miorandi, M. Balboni, S. M. Nowick, and D. Bertozzi, "Accurate assessment of bundled-data asynchronous NoCs enabled by a predictable and efficient hierarchical synthesis flow," in *Proc. IEEE Int. Symp. Asynchronous Circuits Syst.*, 2017, pp. 10–17.
16. J. J. Lecler and G. Baillieu, "Application-driven network-on-chip architecture exploration and refinement for a complex SoC," *Springer Des. Autom. Embedded Syst.*, vol. 15, no. 2, pp. 133–158, 2011.
17. P. Vivet *et al.*, "A 4x4x2 homogeneous scalable 3D network-on-chip circuit with 326 MFlit/s 0.66pJ/bit robust and fault tolerant asynchronous 3D links," *IEEE J. Solid-State Circuits*, vol. 52, no. 1, pp. 33–49, Jan. 2017.
18. M. Gibiluka, M. T. Moreira, F. G. Moraes, and N. L. V. Calazans, "BAT-Hermes: A transition-signaling bundled-data NoC router," in *Proc. IEEE Latin Amer. Symp. Circuits Syst.*, 2015, pp. 1–4.
19. H. F. Tatenguem, D. Ludovici, A. Strano, D. Bertozzi, and H. Reinig, "Contrasting multi-synchronous MPSoC design styles for fine-grained clock domain partitioning: The full-HD video playback case study," in *Proc. ACM Int. Workshop Network-on-Chip Archit.*, 2011, pp. 37–42.
20. Memory bandwidth data, http://nvdla.org/primer.html

DAVIDE BERTOZZI is currently a Professor with the Department of Engineering, University of Ferrara. His research focuses on chip-scale interconnect technology and on its capability to enable new system architectures. Bertozzi has a Ph.D. in electrical engineering from the University of Bologna. He is a member of IEEE, ACM, and the HiPEAC European Network-of-Excellence on High-Performance Embedded Architectures and Compilation. Contact him at brtdvd@unife.it.

GABRIELE MIORANDI is currently a digital circuit designer with the High-Performance Data Center Solutions Group, Microchip, Milan, Italy. Miorandi has a Ph.D. in electrical engineering from the University of Ferrara, where he performed the work for this article. The focus of his research was on architectural optimization and synthesis methods for two-phase bundled-data asynchronous network-on-chip design. Contact him at gabriele.miorandi@gmail.com.

ALBERTO GHIRIBALDI is a co-founder of ArzaMed s.r.l., a web software company in Rimini, Italy, where he is responsible for product research and development, as well as server cloud infrastructure. Ghiribaldi has a Ph.D. in electrical engineering from the University of Ferrara, where he performed the work for this article. The focus of his research was on cost-effective asynchronous NoC switch architectures using two-phase bundled data. Contact him at ghiribaldi.alberto@gmail.com.

WAYNE BURLESON has been a Professor of Electrical and Computer Engineering, University of Massachusetts Amherst, since 1990. From 2012 to 2017, he was a Senior Fellow with AMD Research in Boston, during which he performed the work in this article. His primary research is in the general area of VLSI, including circuits and CAD for clocking and low-power, and security engineering. He has authored more than 200 refereed publications in these areas and is a Fellow of IEEE for contributions in integrated circuit design and signal processing. Contact him at burleson@umass.edu.

GREG SADOWSKI is currently a Technical Fellow with Advanced Micro Devices, Boxborough, MA, USA, where he is involved in the performance optimization of both GPU and ML technologies. His research interests include continued performance and power improvements of machine intelligence and GPU systems. He holds 48 U.S. patents. He is a senior member of IEEE. Contact him at greg.sadowski@amd.com.

KSHITIJ BHARDWAJ is a research staff member at Lawrence Livermore National Laboratory, before which he was a Postdoctoral Research Fellow with the Computer Architecture and VLSI group at Harvard University. His research interests include asynchronous networks-on-chip, GALS systems, many-accelerator SoCs for AI applications, and system-level optimization using machine learning. Bhardwaj has a Ph.D. in computer science from Columbia University, where he performed the work for this article. Contact him at kshitij.b.cs@gmail.com.

WEIWEI JIANG is currently an Architecture and RTL Design Engineer for networking chip development with Google Cloud. His research interests include asynchronous and mixed-timing digital circuits, high-performance and low-power asynchronous NoCs and GALS systems, EDA tools for ASICs and FPGAs. Jiang has a Ph.D. in computer science from Columbia University. He performed the work for this article when interning at AMD Research during his Ph.D. Contact him at ording1985@gmail.com.

STEVEN M. NOWICK is currently a Professor of Computer Science at Columbia University, and a former chair and co-founder of the Computer Engineering Program. He was founding chair of the "Computing Systems for Data-Driven Science" Center in Columbia's Data Science Institute and a co-founder of the IEEE "Async" Symposium. His research interests include asynchronous and GALS digital systems, CAD optimization, scalable high-performance and low-power networks-on-chip, and embedded and ultra-low energy systems. He has collaborated on asynchronous designs with AMD, Boeing, IBM, and NASA Goddard. Nowick has a Ph.D. in computer science from Stanford University. He is a Fellow of IEEE. Contact him at nowick@cs.columbia.edu.