Powering Networks on Chips

Energy-efficient and reliable interconnect design for SoCs

Luca Benini
DEIS Università di Bologna
Bologna, Italy 40136
lbenini@deis.unibo.it

Giovanni De Micheli
CSL Stanford University
Stanford, CA 94305
nanni@stanford.edu

ABSTRACT

We consider *systems on chips* (SoCs) that will be designed and produced in five to ten years from today, with gate lengths in the range 50-100nm. We address the distinguishing features of a design methodology that aims at achieving reliable designs under the limitations of the interconnect technology. Specifically, we consider energy consumption reduction, under guaranteed *quality of service* (QoS), as a main objective in system design.

Keywords

Systems on Chips, low-energy design, networks

1. INTRODUCTION

Systems on chip (SoCs) will be soon designed and fabricated in technologies with gate lengths in the range 50-100nm. Several challenges arise from the complexity of designing billion-transistor chips. Thus current design tools and methods will require evolutionary and revolutionary changes. We believe that such changes will be driven by the following factors:

- Systems on chip will be designed using pre-existing components, such as processors, controllers and memory arrays. Design methodologies will support component re-use in a plug-and-play fashion.
- SoCs will have to provide a functionally-correct, reliable operation of the interacting components. The physical interconnections on chip will be a limiting factor for performance and energy consumption.
- The overall design goal (for most SoCs) will be to satisfy some *quality of service* (QoS) metric with the least energy consumption. Most QoS metrics encompass performance and reliability measures.

In this paper, we address a methodology for the design of energy-efficient communication for SOCs.

ISSS'01, October 1-3, 2001, Montréal, Québec, Canada.

2. TECHNOLOGY TRENDS

The *international technology roadmap for semiconductors* (ITRS) [22] projects that we will be designing multi-billion transistor chips by the end of this decade, with feature sizes around $50nm$ and clock frequencies around $10Ghz$. Delays on wires will dominate: global wires spanning a significant fraction of the chip size will carry signals whose propagation delay will exceed the clock period. Moreover, synchronization of chips with a single clock source and negligible skew will be extremely hard or impossible. The most likely synchronization paradigm for future chips is *globally-asynchronous locally-synchronous* (GALS), with many different clocks. Global wires will span multiple clock domains, and synchronization failures in communicating between different clock domains will be rare but unavoidable events [8].

SoC design will be guided by the principle of consuming the least possible power. This requirement matches the need of using SoCs in portable battery-powered electronic devices and of curtailing thermal dissipation which can make chip operation infeasible or impractical. Energy considerations will impose small logic swings and power supplies, most likely below 1 Volt. Electrical noise due to *cross-talk*, *electro-magnetic interference* (EMI) and radiation-induced charge injection (*soft errors*) will be likely to produce *data upsets*. Thus, the mere transmission of digital values on wires will be *inherently unreliable*.

As a result, the distinguishing challenge for SoC design will be to provide adequate *quality of service* (QoS), with a limited energy budget under strong limitations of the technology. QoS requirements include, but are not limited to, performance and reliability. High performance is demanded by the increasingly complex software applications required to run even on small portable appliances (e.g., multi-media on cell phones). Reliability is mandated by the increasing reliance of consumers on electronic communication and control systems in every day's life.

We propose to use network design technology to analyze and design SoCs. In other words, we view a SoC as a *micro-network* of components. We postulate that SoC interconnect design can be done using the *micro-network stack* paradigm, which is an adaptation of the protocol stack [15] (Figure 1). Thus the electrical, logic, and functional properties of the interconnection scheme can be abstracted.

SoCs differ from wide-area networks because of local proximity and because they exhibit much less non-determinism. Local, high-performance networks (such as those developed for large-scale multiprocessors), have similar requirements and constraints. A few distinctive characteristics are unique

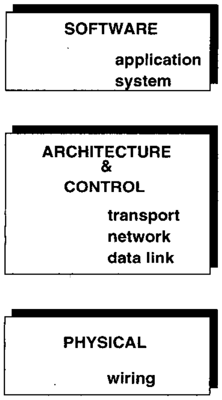

Figure 1: Micro-network stack

of SoC networks, namely, **energy constraints** and **design-time specialization.** This papers addresses specifically the former problem.

The SIA roadmap projects that power consumption can marginally scale up while moving from $100nm$ to $50nm$ technology. At the same time, projected clock frequency and number of devices on-chip are increased significantly. These trends translate directly into much tighter power budgets for all system components. Voltage scaling, as predicted by the roadmap, is helpful in reducing power. Nevertheless, voltage scaling alone will not suffice, and specific design choices for low-energy energy consumption will be required.

Whereas computation and storage energy greatly benefits from device scaling (smaller gates, smaller memory cells), the energy for global communication does not scale down. On the contrary, projections based on current delay optimization techniques for global wires [13, 14, 11] show that global communication on chip will require increasingly higher energy consumption. Hence, communication-energy minimization will be a growing concern in future technologies. Furthermore, network traffic control and monitoring can help in better managing the power consumed by networked computational resources. For instance, clock speed and voltage of end nodes can be varied according to available network bandwidth. The emphasis on energy minimization creates a sleuth of novel challenges that have not been addressed by traditional high-performance network designers.

Design-time specialization is another facet of the SoC network design. Whereas macroscopic networks emphasize general-purpose communication and modularity, in SoCs networks these constraints are less restrictive. The communication network fabric is designed on silicon from scratch. Standardization is needed only for specifying an abstract network interface for the end nodes, but the network architecture itself can be tailored to the application, or class of applications, targeted by the SoC design. Hence, we envision a vertical design flow where every layer of the micro-network stack is specialized and optimized for the target application domain. From a design stand-point the network reconfigurability will be key in providing plug-and-play use of components, since they will interact with the others through (reconfigurable) protocols.

3. A LAYERED DESIGN METHODOLOGY

We consider now on-chip communication and its abstraction as a micro-network. We analyze the various levels of the micro-network stack bottom-up.

3.1 Physical layer

Global wires are the physical implementation of the communication channels. Physical layer signaling techniques for lossy transmission lines have been studied for a long time by high-speed board designers and microwave engineers [2, 8].

Traditional rail-to-rail voltage signaling with capacitive termination, as used today for on-chip communication, is definitely not well-suited for high-speed, low-energy communication on future global interconnect [8]. Reduced swing, current-mode transmission, as used in some processor-memory systems, can significantly reduce communication power dissipation while preserving speed of data communication.

Nevertheless, as the technology trends lead us to use smaller voltage swings and capacitances, the upset probabilities will rise. Thus the trend toward faster and lower-power communication may decrease reliability as an unfortunate side effect. Reliability bounds as voltages scale can be derived from theoretical (entropic) considerations [5] and can be measured also by experiments on real circuits.

We conjecture that a paradigm shift is needed to address the aforementioned challenges. Current design styles consider wiring-related effects as undesirable parasitics, and try to reduce or cancel them by specific and detailed physical design techniques. It is important to realize that a well-balanced design should not over-design wires so that their behavior approaches an ideal one, because the corresponding cost in performance, energy-efficiency and modularity may be too high. Physical layer design should find a compromise between competing quality metrics and provide a clean and complete abstraction of channel characteristics to micro-network layers above.

3.2 Data link, network and transport layers

Due to the limitations at the physical level and to the high bandwidth requirement, we conjecture that SoC design will use network architectures similar to those used for multi-processors. Whereas *shared medium* (e.g., bus-based) communication dominates in today's chip designs, scalability reasons make us conjecture that more general network topologies will be used in the future. In this perspective, micro-network design entails the specification of *network architectures* and *control protocols* [9]. The architecture specifies the topology and physical organization of the interconnection network, while the protocols specify how to use network resources during system operation.

The *data-link layer* abstracts the physical layer as an unreliable digital link, where the probability of bit upsets is non null (and increasing as technology scales down). Furthermore, reliability can be traded off for energy [5]. The main purpose of data-link protocols is to increase the reliability of the link up to a minimum required level, under the assumption that the physical layer by itself is not sufficiently reliable.

An additional source of errors is contention in shared-medium networks. Contention resolution is fundamentally a non-deterministic process, because it requires synchronization of a distributed system, and for this reason it can be seen as an additional noise source. In general, non determinism can be virtually eliminated at the price of some performance penalty. For instance, centralized bus arbitration in a synchronous bus eliminates contention-induced errors, at the price of a substantial performance penalty caused by the slow bus clock and by bus request/release cycles.

Future high-performance shared-medium on-chip micronetworks may evolve in the same direction as high-speed local area networks, where contention for a shared communication channel can cause errors, because two or more transmitters are allowed to concurrently send data on a shared medium. In this case, provisions must be made for dealing with contention-induced errors.

An effective way to deal with errors in communication is to *packetize* data. If data is sent on an unreliable channel in packets, error containment and recovery is easier, because the effect of errors is contained by packet boundaries, and error recovery can be carried out on a packet-by-packet basis. At the data link layer, error correction can be achieved by using standard *error correcting codes* (ECC) that add redundancy to the transferred information. Error correction can be complemented by several packet-based error detection and recovery protocols. Several parameters in these protocols (e.g., packet size, number of outstanding packets, etc.) can be adjusted depending on the goal to achieve maximum performance at a specified residual error probability and/or within given energy consumption bounds.

At the *network layer*, packetized data transmission can be customized by the choice of switching and routing algorithms. The former, (e.g., *circuit, packet*, and *cut-through* switching), establish the type of connection while the latter determine the path followed by a message through the network to its final destination. Switching and routing for on-chip micro-networks affect heavily performance and energy consumption. We conjecture that future approaches will emphasize speed and decentralization of routing decisions. Robustness and fault tolerance will also be highly desirable.

At the *transport layer*, algorithms deal with the decomposition of messages into packets at the source and their assembly at destination. Packetization granularity is a critical design decision, because the behavior of most network control algorithms is very sensitive to packet size. Packet size can be application-specific in SoCs, as opposed to general networks. In general, flow control and negotiation can be based on either deterministic or statistical procedures. Deterministic approaches ensure that traffic meets specifications, and provide hard bounds on delays or message losses. The main disadvantage of deterministic techniques is that they are based on worst cases, and they generally lead to significant under-utilization of network resources. Statistical techniques are more efficient in terms of utilization, but they cannot provide worst case guarantees. Similarly, from an energy viewpoint, we expect deterministic schemes to be more inefficient than statistical schemes, because of their implicit worst-case assumptions.

3.3 Software layers

Software layers comprise system and application software. We limit our considerations to system software, which includes processing element and network operating systems. The system software provides us with an abstraction of the underlying hardware platform. In a nutshell, we can view the system as a queueing network of components. Each component models a computational or storage unit, while the queueing network abstracts the micro-network. Moreover, we can assume that:

- Each component can operate at various service levels, providing corresponding performance and energy consumption levels. This abstracts the physical implementation of components with adjustable voltage and/or frequency levels, as well as with the ability to disable their functions in full or in part.
- The information flow between the various units can be controlled by the system software to provide the appropriate quality of service. This entails controlling the routing of the information, the local buffering into storage arrays and the rate of the information flow.

The system software must support *dynamic power management* (DPM) of its components as well as *dynamic information-flow management.* DPM is currently applied within system design. We consider dynamic voltage/frequency scaling as a form of power management, because its limiting cases, i.e., powering off or stopping the clock, are the most common realizations of DPM.

Dynamic power management entails selecting the appropriate component state to service a workload with the minimum energy consumption. DPM *policies* are the control algorithms for state transitions [3]. Note that transitions among states have a finitê delay penalty, even when changing the operation frequency. Thus, the computation of policies that maximize performance under energy constraints, or solving the dual problem, may be computationally complex [3, 4].

Dynamic information-flow management relates to configuring the network and the bandwidth of the local interconnection to satisfy the information flow requirements. This problem is tightly related to DPM, and can be seen as an application of DPM to the micro-network instead of to a component. Again, policies implemented at the system software layer request either specific protocols or parameters at the lower layers to achieve the appropriate information flow, using the least amount of resources and energy.

4. ENERGY-EFFICIENT DESIGN: TECHNIQUES AND EXAMPLES

In this section we delve in a few specific instances of energy-efficient micronetwork design problems. In most cases, we also outline specific solutions that have been proposed in the literature, even though it should be clear that many design issues are open and significant progress in this area is expected in the near future.

4.1 Physical layer

At the physical layer, low-swing signaling is actively investigated to reduce communication energy on global interconnects [20]. In the case of a simple CMOS driver, low-swing signaling is achieved by lowering the driver's supply voltage V_{dd}. This implies a quadratic dynamic power reduction (because $P_{dyn} = KV_{dd}^2$). Unfortunately, swing reduction at the transmitter complicates the receiver's design. Increased sensitivity and noise immunity are required to guarantee reliable data reception. Differential receivers have superior sensitivity and robustness, but they require doubling the bus width. To reduce the overhead, pseudo-differential schemes have been proposed, where a *reference* signal is shared among several bus lines and receivers, and incoming data is compared against the reference in each receiver. Pseudo-differential signaling reduces the number of signal transitions, but it has reduced noise margins with respect to fully differential signaling. Thus, reduced switching activity is counterbalanced by higher swings and determining the minimum-energy solution requires careful circuit-level analysis.

Another key physical-layer issue is synchronization. Traditional on-chip communication has been based on the synchronous assumption, which implies the presence of global synchronization signals (*i.e.*, clocks) that define data sampling instants throughout the chip. Unfortunately, clocks are extremely energy-inefficient, and it is a well-known fact that they are responsible for a significant fraction of the power budget in digital integrated systems. Thus, postulating global synchronization when designing on-chip micronetworks is not an optimal choice from the energy viewpoint. Alternative on-chip synchronization protocols that do not require the presence of a global clock have been proposed in the past [21, 23] but their effectiveness has not been studied in detail from the energy viewpoint.

4.2 Data-link layer

At the data-link layer, a key challenge is to achieve the specified communication reliability level with minimum energy expense. Several error recovery mechanisms developed for macroscopic networks can be deployed in on-chip micronetworks, but their energy efficiency should be carefully assessed in this context. As a practical example, consider two alternative reliability-enhancement techniques: *error-correcting codes* and *error-detecting codes with retransmission.* Both approaches are based on transmitting redundant information over the data link, but error-correction is generally more demanding than error detection in terms of redundancy and decoding complexity. Hence, we can expect error-correcting transmission to be more power-hungry in the error-free case. However, when an error arises, error detecting schemes require retransmission of the corrupted data. Depending on the network architecture, retransmission can be very costly in terms of energy (and performance).

Clearly, the tradeoff between the increased cost of error correction and the energy penalty of retransmission should be carefully explored when designing energy-efficient micronetworks [5]. Either scheme may be optimal, depending on system constraints and on physical channel characteristics. Automatic design space exploration could be very beneficial in this area.

In case of shared-medium network links (such as busses), the media-access-control function of the data link layer is also critical for energy efficiency. Currently, centralized time-division multiplexing schemes (also called centralized arbitration) are widely adopted [1, 7, 17]. In these schemes, a single arbiter circuit decides which transmitter accesses to the bus for every time slot. Unfortunately, the poor scalability of centralized arbitration indicates that this approach is likely to be energy-inefficient as micronetwork complexity scales up. In fact, the energy cost of communicating with the arbiter, and hardware complexity of the arbiter itself scale up more than linearly with the number of bus masters

Distributed arbitration schemes as well as alternative multiplexing approaches, such as code division multiplexing, have been extensively adopted in shared-medium macroscopic network, and are actively investigated for on-chip communication [18]. However, research in this area just burgeoning, and significant work is needed to develop energy-aware media-access-control for future micronetworks.

4.3 Network layer

Network architecture heavily influences communication energy. As hinted in the previous section, shared-medium networks (busses) are currently the most common choice, but it is intuitively clear that busses are not energy-efficient as network size scales up [10]. In bus-based communication, data is always broadcasted from one transmitter to all possible receivers, while in most cases messages are destined to only one receiver, or a small group. Bus contention, with the related arbitration overhead further contributes to the energy overhead.

Preliminary studies on energy-efficient on-chip communication indicate that hierarchical and heterogeneous architectures are much more energy-efficient than busses [12, 21]. In their work, Zhang *et al.* [21] develop a *hierarchical generalized mesh* where network nodes which high communication bandwidth requirement are clustered and connected through a programmable generalized mesh consisting of several short communication channels joined by programmable switches. Clusters are then connected through a generalized mesh of global long communication channel. Clearly such architecture is heterogeneous because the energy cost of intra-cluster communication is much smaller than that of inter-cluster communication. While the work of Zhang *et al.* demonstrates that power can be saved by optimizing network architecture, many network design issues are still open, and we need tools and algorithms to explore the design space and to tailor network architecture to specific applications or classes of applications.

Network architecture is only one facet of network layer design, the other major facet being network control. A critical issue in this area is the choice of a switching scheme for indirect network architectures. From the energy viewpoint, the tradeoff is between the cost of setting up a circuit-switched connection once for all, and the overhead of switching packets throughout the entire communication time on a packet-based connection. In the former case the network control overhead is "lumped" and incurred once, while in the latter case, it is distributed over many small contributions, one for each packet. When communication flow between network nodes is extremely persistent and stationary, circuit-switched schemes are likely to be preferable, while packet switched schemes should be more energy-efficient for irregular and non-stationary communication patterns. Needless to say, circuit switching and packet switching are just two extremes of a spectrum, with many hybrid solutions in between [15].

4.4 Transport layer

Above the network layer, the communication abstraction is an end-to-end connection. The transport layer is concerned with optimizing the usage of network resources and providing a requested quality of service. Clearly, energy can be seen as a network resource or a component in a quality-of-service metric. An example of transport-layer design issue is the choice between connection-oriented and connection-less protocols. Energy efficiency can be heavily impacted by this decision. In fact, connection-oriented protocols can be energy inefficient under heavy traffic conditions because they tend to increase the number of re-transmissions. On the other hand, out-of-order delivery of data may imply additional work at the receiver, which causes additional energy consumption. Thus, communication energy should be balanced against computation energy at destination nodes.

Another transport-layer task with far-reaching implications on energy is flow control. When many transmitter compete for limited communication resources, the network becomes congested and the cost per transmitted bit increases, because of increased contention and contention resolution overhead. Flow control can mitigate the effect of congestion by regulating the amount of data that enters the network, at the price of some throughput penalty. Energy reduction by flow control has been extensively studied for wireless networks [15, 16], but it is an unexplored research area for on-chip micronetworks.

4.5 Application and system layer

End-nodes (processing elements) in SoC micronetworks will most likely be power-manageable, and one of the key tasks of the system software will be to control their power states. We can envision two alternative approaches to the power management problem, namely *node-centric* and *network-centric*. In the former case, system software running on a power-manageable component determines its state transitions, based on the system state and on its workload. Thus, the system software of the component has a local DPM policy and controls the underlying hardware through appropriate system calls. In the latter case, components send messages to neighbors to request state changes. Such requests originate and are serviced at the system software levels. For example, an image processor can be required to raise its service levels before receiving a stream of data. In this case, the system software supports policies that accept requests from other components and perform transitions according to such requests. At the same time, the policies can originate requests for other components. We conjecture that system-level power management for future SoCs will gradually evolve from a node-centric toward a network-centric view.

Application software development for SoCs should achieve two major goals. First, preserve portability and generality of the applications across different platforms. Second, provide some intelligence to leverage the distributed nature of the underlying platform for reducing its energy consumption. These two objectives are apparently conflicting. A possible strategy to satisfy both goals is to provide power-aware *application programming interfaces* (APIs) that let applications dialogue with system software. Thus, applications may acquire information from the system software about the specific platform. At the same time, applications can request the system software to set the hardware in specific power states to provide adequate quality of service with minimal power. System software can serve or deny requests from applications, according to the state of other running processes. The design of energy-aware APIs is an open and promising research area.

5. CONCLUSIONS

The challenges of designing SoCs in 50-100nm technologies available in the second part of this decade include coping with design complexity, providing reliable, high-performance operation and minimizing energy consumption. Starting from the observation that interconnect technology will be the limiting factor for achieving the operational goals, we envisioned a communication-centric view of design. We focused on energy efficiency issues in designing the communication infrastructure for future SoCs. We described several open problems at various layers of the communication stack, and we outlined basic strategies to effectively tackle the energy efficiency challenge for on-chip communication networks.

6. REFERENCES

[1] P. Aldworth, "System-on-a-Chip Bus Architecture for Embedded Applications," *IEEE International Conference on Computer Design*, pp. 297-298, 1999.

[2] H. Bakoglu, *Circuits, Interconnections, and Packaging for VLSI*, Addison-Wesley, 1990.

[3] L. Benini, A. Bogliolo, G. De Micheli, "A Survey of Design Techniques for System-Level Dynamic Power Management," *IEEE Transactions on Very Large-Scale Integration Systems*, vol. 8, no. 3, pp. 299-316, June 2000.

[4] L. Benini, G. De Micheli, "System-Level Power Optimization: Techniques and Tools," *ACM Transactions on Design Automation of Electronic Systems*, vol. 5, no. 2, pp. 115-192, April 2000.

[5] R. Hegde, N. Shanbhag, "Toward achieving energy efficiency in presence of deep submicron noise," *IEEE Transactions on VLSI Systems*, pp. 379–391, vol. 8, no. 4, August 2000.

[6] D. Bertsekas, R. Gallager, *Data Networks.* Prentice Hall, 1991.

[7] B. Cordan, "An efficient bus architecture for system-on-chip design," *IEEE Custom Integrated Circuits Conference*, pp. 623–626, 1999.

[8] W. Dally and J. Poulton, *Digital Systems Engineering*, Cambridge University Press, 1998.

[9] J. Duato, S. Yalamanchili, L. Ni, *Interconnection Networks: an Engineering Approach.* IEEE Computer Society Press, 1997.

[10] P. Guerrier, A. Grenier, "A generic architecture for on-chip packet-switched interconnections," *Design Automation and Test in Europe Conference*, pp. 250–256, 2000.

[11] R. Ho, K. Mai, M. Horowitz, "The Future of wires," *Proceedings of the IEEE*, January 2001.

[12] C. Patel, S. Chai, S. Yalamanchili, D. Shimmel, "Power constrained design of multiprocessor interconnection networks," *IEEE International Conference on Computer Design*, pp. 408-416, 1997.

[13] D.Sylvester and K.Keutzer, "A Global Wiring Paradigm for Deep Submicron Design," *IEEE Transactions on CAD/ICAS*, Vol.19, No. 2, pp. 242-252, February 2000.

[14] T. Theis, "The future of Interconnection Technology," *IBM Journal of Research and Development*, Vol. 44, No. 3, May 2000, pp. 379-390.

[15] J. Walrand, P. Varaiya, *High-Performance Communication Networks*. Morgan Kaufman, 2000.

[16] I. Papadimitriou, M. Paterakis, "Energy-conserving access protocols for transmitting data in unicast and broadcast mode," *International Symposium on Personal, Indoor and Mobile Radio Communication*, pp. 416–420, 2000.

[17] S. Winegarden, "A bus architecture centric configurable processor system," *IEEE Custom Integrated Circuits Conference*, pp. 627–630, 1999.

[18] R. Yoshimura, T. Koat, S. Hatanaka, T. Matsuoka, K. Taniguchi, "DS-CDMA wired bus with simple interconnection topology for parallel processing system LSIs," *IEEE Solid-State Circuits Conference*, pp. 371-371, Jan. 2000.

[19] H. Zhang, V. Prabhu, V. George, M. Wan, M. Benes, A. Abnous, J. Rabaey, "A 1-V Heterogeneous Reconfigurable DSP IC for Wireless Baseband Digital Signal Processing," *IEEE Journal of Solid-State Circuits*, vol. 35, no. 11, pp. 1697–1704, Nov. 2000.

[20] H. Zhang, V. George, J. Rabaey, "Low-swing on-chip signaling techniques: effectiveness and robustness," *IEEE Transactions on VLSI Systems*, vol. 8, no. 3, pp. 264-272, June 2000.

[21] H. Zhang, M. Wan, V. George, J. Rabaey, "Interconnect architecture exploration for low-energy configurable single-chip DSPs," *IEEE Computer Society Workshop on VLSI*, pp. 2-8, 1999.

[22] International Technology Roadmap for Semiconductors *http://public.itrs.net/*

[23] W. Bainbridge, S. Furber, "Delay insensitive system-on-chip interconnect using 1-of-4 data encoding," *IEEE International Symposium on Asynchronous Circuits and Systems*, pp. 118-126, 2001.