Chapter 12

ENERGY-EFFICIENT NETWORK-ON-CHIP DESIGN

Low Power NoC Design Techniques

Davide Bertozzi
University of Bologna
dbertozzi@deis.unibo.it

Luca Benini
University of Bologna
lbenini@deis.unibo.it

Giovanni De Micheli
Stanford University
nanni@stanford.edu

Abstract Performance and power consumption of multi-processor Systems-on-Chip (SoCs) are increasingly determined by the scalability properties of the on-chip communication architecture. Networks-on-Chip (NoCs) are a promising solution for efficient interconnection of SoC components. This chapter focuses on low power NoC design techniques, analyzing the related issues at different layers of abstraction and providing examples taken from the most advanced NoC implementations presented in the open literature. Particular emphasis is given to application-specific NoC architectures, in that they represent the most promising scenario for minimization of communication-energy in multi-processor SoCs.

Keywords: Network-on-Chip, Low Power, Micro-network Stack, Application-Specific

Introduction

The most critical factor in *Systems-on-Chip* (SoCs) integration will be related to the communication scheme among components. The challenges for on-chip interconnect stem from the physical properties of the interconnection wires. Global wires will carry signals whose propagation delay will exceed the clock period. Thus signals on global wires will be pipelined. At the same time, the switched capacitance on global wires will constitute a significant fraction of the dynamic power dissipation. Moreover, estimating delays accurately will become increasingly harder, as wire geometries may be determined late in the design flow. Hence, the need for latency insensitive design is critical. The most likely synchronization paradigm for future chips is *globally-asynchronous locally-synchronous* (GALS), with many different clocks.

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. 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 [20] show that global on-chip communication will require increasingly higher energy consumption. Hence, communication-energy minimization will be a growing concern in future technologies [40].

Energy considerations will impose small logic swings and power supplies, most likely below 1 Volt. Electrical noise due to *cross-talk*, *electromagnetic 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.*

To cope with these problems, network design technology can be used to analyze and design SoCs modeled as *micro-networks* of components (or *Networks-on-Chip*, NoCs). The SoC interconnect design analysis and synthesis is based upon the *micro-network stack* paradigm, which is an *adaptation* of the protocol stack [25] (Figure 12.1) used in networking. This abstraction is useful for layering micro-network protocols and separating design issues belonging to different domains.

SoCs differ from wide-area networks because of local proximity and because they exhibit much less non-determinism. In particular, micro-networks have a few distinctive characteristics, namely, *energy constraints*, *design-time specialization* and *low communication latency.*

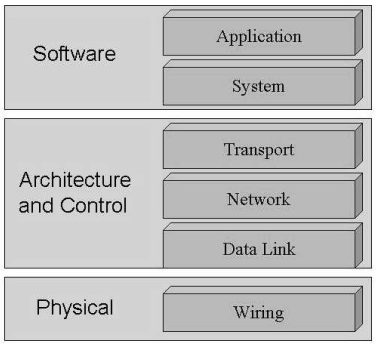


Figure 12.1. Micro-network stack

This chapter focuses on low power NoC design techniques, and analyzes specific design issues related to the different layers of abstraction outlined in the micro-network stack in a bottom-up way. The objective is to describe, for each layer, how the system interconnect is progressively abstracted and what the most relevant micro-network design issues are in order to come up with an energy-efficient NoC architecture. Particular emphasis is given to customized, domain-specific NoCs, which represent the most promising scenario for communication-energy minimization in the context of NoC-based *multi-processor SoCs (MPSoCs)*. In most cases, specific solutions proposed in the literature are outlined, even though it should be clear that many design issues are open and significant progress in this area is expected in the near future.

12.1 Physical layer

Global wires are the physical implementation of the communication channels. 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 [16]. Reduced swing, current-mode transmission, as used in some processor-memory systems, can significantly reduce communication power dissipation while preserving speed of data communication [29].

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 robust-

ness, 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 also 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.

Dynamic voltage scaling has been recently applied to busses [23, 26]. In [26] the voltage swing on communication busses is reduced, even though signal integrity is partially compromised. Encoding techniques can be used to detect corrupted data which is then retransmitted. The retransmission rate is an input to a closed-loop DVS control scheme, which sets the voltage swing at a trade-off point between energy saving and latency penalty (due to data retransmission).

The *On-Chip Network (OCN)* for low power heterogeneous SoC platforms illustrated in [9] employs some advanced techniques for low-power physical interconnect design. OCN consists of global links connecting clusters of tightly-connected IPs which are several millimeters long. By using overdrivers, clocked sense amplifiers and twisted differential signaling, packets are transmitted reliably with less than 600 mV swing. The size of a transceiver and the overdrive voltage are chosen to obtain a 200 mV separation at the receiver end. A 5 mm global link of 1.6um wire-pitch can carry a packet at 1.6GHz with 320ps wire-delay and consumes 0.35pJ/bit. On the contrary, a full-swing link consumes up to 3x more power and additional area of repeaters.

An on-chip serialization technique [10] is also used in OCN, thus significantly reducing area. However, the number of signal transitions on a link is increased since the temporal locality between adjacent packets is removed. An ad-hoc serialized low-energy transmission coding scheme was therefore designed as an attempt to exploit temporal locality between packets. The encoder generates a '1' only when there is difference between a current packet and a previous packet before it is serialized. The decoder then uses this encoded packet to reconstruct the original input, using its previously stored packet. A 13.4% power saving is obtained for a multimedia application. The power overhead associated with the encoder/decoder is only 0.4mW.

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 [19] and can be measured also by experiments on real circuits.

Finally, another key physical-layer issue is synchronization. In fact, global synchronization signals (*i.e.*, clocks) are responsible for a significant fraction of the power budget in digital integrated systems. Alternative on-chip synchronization protocols that do not require the presence of a global clock have been proposed in the past [30, 11] but their effectiveness has not been studied in detail from the energy viewpoint.

In the OCN NoC [9], a programmable power management unit provides four clocks with PLL; 1.6GHz for the OCN, 800 MHz for schedulers, 100MHz for processors and 50 MHz for peripherals. Those clock frequencies are scalable by software for power-mode control and also for optimal operation of each application.

12.2 System Interconnect Architecture

Designing the architecture for an on-chip interconnect requires choices at a higher level of abstraction with respect to physical interconnect design, but also with a stronger impact on energy dissipation.

Traditional shared buses try to overcome their energy inefficiency by means of bus splitting [5]. The bus is split into smaller segments and proper bridges are inserted to ensure communication between any two adjacent segments when needed. Thus, the load capacitance charged and discharged at each bus access is reduced. Most commercial shared buses make use of this solution, including AMBA bus [1] and IBM CoreConnect [33]. They split the bus based on the characteristics of the connected masters and slaves (e.g. high performance cores versus slow peripherals). More advanced bus specifications (such as AMBA Multi-Layer [34]) allow to group IP cores into clusters, and this can be done based on their interaction during application execution.

As an example, OCN [9] exploits locality of IP cores by grouping them into clusters, and a crossbar switch is used for intra-cluster packets, performing buffer-less cut-through switching. A round-robin scheduling of the switch ensures fairness and starvation-freedom to OCN. An $n \times n$ crossbar fabric comprises n^2 crosspoint junctions which contain NMOS pass-transistors. In a conventional crossbar fabric, each input driver wastes power to charge two long wires (horizontal and vertical) and $2n$ transistor-junction-capacitors. OCN employs a crossbar partial activation technique. By splitting the crossbar fabric into 4×4 tiles, input and output wires can be divided into four. A gated input driver at each tile is activated only when the scheduler grants access to the tile. The output signal does not propagate to other tiles to reduce the power con-

sumption on the vertical wire. A 43% power saving is obtained in a 16×16 crossbar switch fabric with a negligible area overhead.

Network topology

Energy considerations might affect the on-chip network topology selection process, as showed by the architectural choices made in the design of recently proposed NoC solutions.

Again, the OCN case is very instructive. In fact, a star topology guaranteeing constant and minimum switch hop counts between every communicating IP was adopted in an early implementation [10]. However, a 1-level flat star topology results in a number of capacitive global wires that may cause long latency and large power dissipation. Therefore, the most recent solution consists of a hierarchical SoC composed of clusters of tightly, star-connected IPs.

Octagon [21] on-chip communication architecture consists of 8 nodes and 12 bi-directional links connected according to an octagonal topology. In this way, communication between any pairs of nodes can be performed by at most two hops. Moreover, Octagon exhibits higher aggregate throughput than a shared bus or crossbar interconnect, a simple, shortest past routing algorithm and less wiring than a crossbar interconnect. Octagon and OCN are examples of network topologies that try to provide the highest degree of connectivity between network nodes while trying to minimize the number of hops, therefore targeting high-performance and low-power NoC realizations.

Power-aware topology selection is briefly discussed in [31] with respect to the SoCIN NoC architecture. A mesh topology is compared with a torus one: the former exhibits lower costs, while the latter reduces message latency. To avoid the long wrapping-around links, with a very high associated capacitive load, a folded torus topology can be used [13]. Such approach reduces the wiring lengths and the power consumption while allowing to improve the operating frequency of network channels.

A more detailed comparison between the power efficiency of a mesh and a folded torus topology is addressed in [13]. The power has been decomposed into the power per hop (traversal of input and output controllers) and power per wire distance travelled. The analysis shows that if wire transmission power dominates per-hop power, the mesh is more power efficient. For the 16 tile network considered in [13], the wire transmission power was estimated to be significantly greater than per-hop power, however the power overhead of the torus was small (less than 15%), and was counterbalanced by the benefits of its larger effective bandwidth.

12.3 Data link layer

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 [19]. 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 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.

For the realization of on-chip micro-networks, several error recovery mechanisms developed for macroscopic networks can be deployed, 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.* A set of experiments involved applying error correcting and detecting codes to an AMBA bus and comparing the energy consumption in four cases [12]: 1) original unencoded data; 2) single-error correction, 3) single-error correction and double-error detection, 4) multiple-error detection. Hamming codes were used. Note that in case 3, a detected double error requires retransmission. In case 4, using (n, k) linear codes, $2^n - 2^k$ errors patterns of length n can be detected. In all cases, some errors may go undetected and be catastrophic. Using the property of the codes, it is possible to map the *mean time to failure* (MTTF) requirement into bit upset probabilities, and thus comparing the effectiveness of the encoding scheme in a given noisy channel (characterized by the upset probability) in meeting the MTTF target.

The energy efficiency of various encoding schemes varies: we summarize here one interesting case, where three assumptions apply. First, wires are long enough so that the corresponding energy dissipation dominates encoding/decoding energy. Second, voltage swing can be lowered until the MTTF target is met. Third, upset probabilities are computed using a white Gaussian noise model [18]. Figure 12.2 shows the average energy per useful bit as a function of the MTTF (which is the inverse of the residual word error probability). In particular, for reliable SoCs (i.e., for MTTF = 1 year), multiple-error detection with retransmission is shown to be more efficient than error-correcting schemes. We refer the reader to [12] for results under different assumptions.

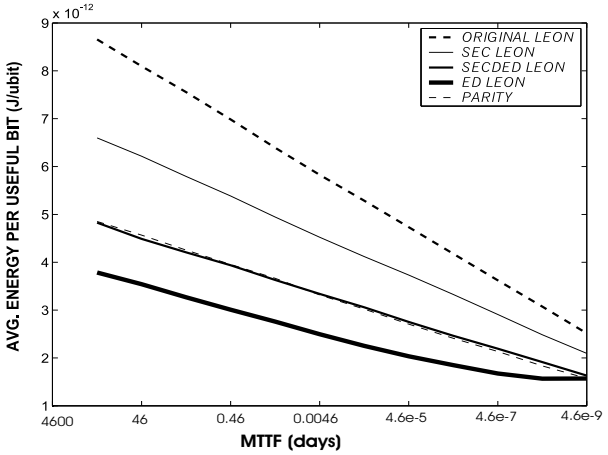


Figure 12.2. Energy efficiency for various encoding schemes

Another important aspect affecting the energy consumption is the *media access control* (MAC) function. Currently, centralized arbitration schemes are widely adopted [1, 14] for the serialization of bus access requests. Unfortunately, central arbiters are instance-specific and therefore poorly scalable. 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. The selection of a specific arbitration algorithm impacts both performance and power consumption [32]. Alternative multiplexing approaches, such as code division multiplexing, are actively investigated for on-chip communication [27]. However, research in this area is just burgeoning, and significant work is needed to develop energy-aware media-access-control for future micro-networks.

Arbitration mechanisms are required also in the implementation of NoC switches to address contention resolution problems such as: prioritizing one out of multiple input channels whose packets have to be directed to the same output channel or multiplexing multiple virtual channels onto the same physical output link.

12.4 Network layer

At the *network layer*, packetized data transmission can be customized by the choice of switching and routing algorithms. The former establishes the type of connection while the latter determines the path followed by a message through the network to its final destination.

In *circuit switching* data and control are separated: control is provided to the network just to set up a connection over which all subsequent data is transported in a connection-free fashion.

On the contrary, *packet-switched* on-chip networks naturally offer best effort services, as contention takes place at the granularity of individual packets. Packet arrival cannot be predicted and contention has to be resolved dynamically: power-hungry data storage is required at the routers for this purpose, and the provision of guarantees is complicated. However, a better link utilization is achieved and error control is made easier.

The most promising packet switching technique for NoC application is *wormhole switching*. It was originally designed for parallel computer clusters [17] because it achieves the minimal network delay and requires fewer buffers. In wormhole switching, each packet is further segmented into *flits* (flow control unit). The header flit reserves the routing channel of each switch, the body flits will then follow the reserved channel, the tail flit will later release the channel reservation.

One major advantage of wormhole switching is that it does not require the complete packet to be stored in the switch while waiting for the header flit to be routed to the next stages. Wormhole switching not only reduces the store-and-forward delay at each switch, but it also requires much less buffer spaces. Because of these advantages, wormhole switching is an ideal candidate technique for on-chip interconnect networks [13], although *deadlock* and *livelock* are potential problems that need to be taken care of [17, 15].

Routing algorithms can be static (packets injected into the network already include routing information and only minimum header processing is required at the switches) or dynamic (routing decisions are dynamically taken at the switches). These latter policies allow packet routes to adapt to network conditions, and therefore trade-off the energy savings obtained in this way with the increased switch complexity and related energy dissipation. Next, a comparison between energy efficiency of routing techniques is provided as an example of network-level design decisions for low power.

Contention-Look-Ahead Routing A *contention-look-ahead* routing scheme is the one where the current routing decision is helped by monitoring the adjacent switches, thus possibly avoiding or reducing blockages and contention in the coming stages.

A contention-aware routing scheme is described in [22]. The routing decision at every node is based on the "stress values" (the traffic loads of the neighbors) that are propagated between neighboring nodes. This

scheme is effective in avoiding "hot spots" in the network. The routing decision steers the packets to less congested nodes.

To solve the contention problems in *wormhole switching* schemes, a contention-look-ahead routing algorithm can be used, which "foresees" the contention and delays in the coming stages using a direct connection from the neighboring nodes. The major difference from [22] is that information is handled in flits, and thus large and/or variable size packets can be handled with limited input buffers. Furthermore, because it avoids contention between packets and requires much less buffer usage, the latter contention-look-ahead routing scheme can greatly reduce the network power consumption.

At every intermediate stage, there may be many alternate routes to go to the next stage. We call the route that always leads the packet closer to the destination a *profitable route*. Conversely, a route that leads the packet away from the destination is called *misroute* [17]. In mesh networks, profitable routes and misroutes can be distinguished by comparing the current node ID with the destination node ID.

Profitable routes will guarantee a shortest path from source to destination. Nevertheless misroutes do not necessarily need to be avoided. Occasionally, the buffer queues in all available profitable routes are full, or the queues are too long. Thus detouring to a misroute may lead to a shorter delay time. Under these circumstances, a misroute may be more desirable.

It is interesting to compare the contention-look-ahead routing algorithm with *dimension order routing* – a routing scheme that always routes the packets on one dimension first, upon reaching the destination row or column, then switches to the other dimension until reaching the destination. Dimension ordered routing is deterministic and guarantees shortest path, but it cannot avoid contention.

The contention-look-ahead routing will reduce the power consumption on the buffers because it can "foresee" the contention in the forthcoming stages and shorten the buffer queue length. On the contrary, dimension-ordered routing always steers the packets along the shortest path, while contention-look-ahead routing may choose the misroute when contention occurs and therefore has a larger average hop count per packet. This translates to more power on the interconnect.

Finally, the contention-look-ahead routing switch needs more logic gates than dimension-ordered routing. However, simulation results show that with 16 RISC processors on a $4x4$ mesh interconnect, contention-look-ahead routing reduces the total network power by about 15% with 16-flit buffers. The reduction is more significant with larger buffer sizes.

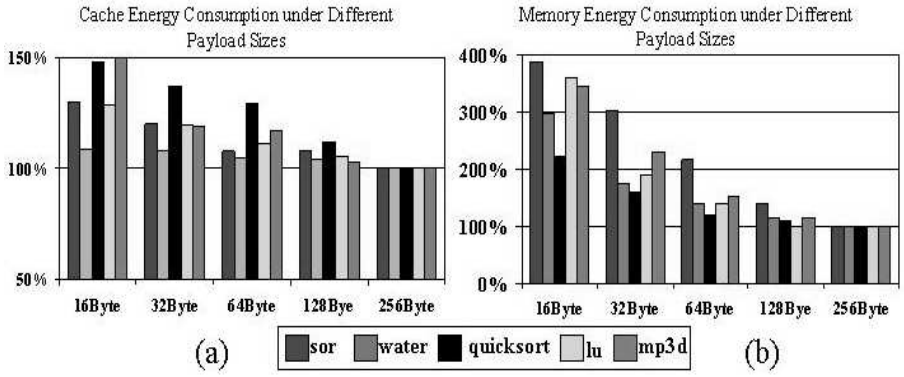


Figure 12.3. Cache and Memory Energy Decrease as Packet Payload Size Increases

12.5 Transport layer

At the *transport layer*, algorithms deal with the decomposition of messages into packets at the source and their assembly at destination. The choice of information decomposition into packets or flits, as well as the choice of packet size can heavily impact energy efficiency. Next, we will use the shared-memory MPSoC as a case study to analyze the packet size trade-offs both qualitatively and quantitatively. The system architecture consists of an on-chip interconnect which provides connectivity to nodes composed by a RISC processor, its caches, a local memory reachable by means of a local bus and the network interface. The MPSoC power consumption originates from three sources: 1) the node processor power consumption, 2) the cache and shared memory power consumption, and 3) the interconnect network power consumption. We will start first from the cache and memory analysis.

Cache and memory power consumption Whenever there is a cache miss, the cache block content needs to be encapsulated inside the packet payload and sent across the network. In shared-memory MPSoC, the cache block size correlates with the packet payload size. Larger packet sizes will decrease the cache miss rate, because more cache content can be updated in one memory access. Consequently, both cache energy consumption and memory energy consumption will be reduced. This relationship can be seen from Fig. 12.3. It shows the energy consumption of cache and memory under different packet sizes. The energy in the figure is normalized to the value of 256Byte, which achieves the minimum energy consumption.

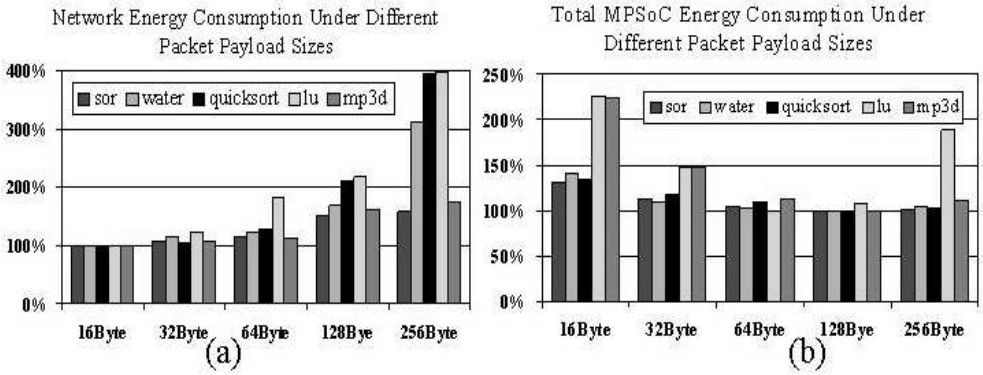


Figure 12.4. Network and Total MPSoC Energy Consumption under Different Packet Payload Sizes

Interconnect network power consumption The power consumption of packetized dataflow on MPSoC network is determined by the following three factors: 1) the number of packets on the network, 2) the energy consumed by each packet on one hop, and 3) the number of hops each packet travels. We summarize these effects and list them below:

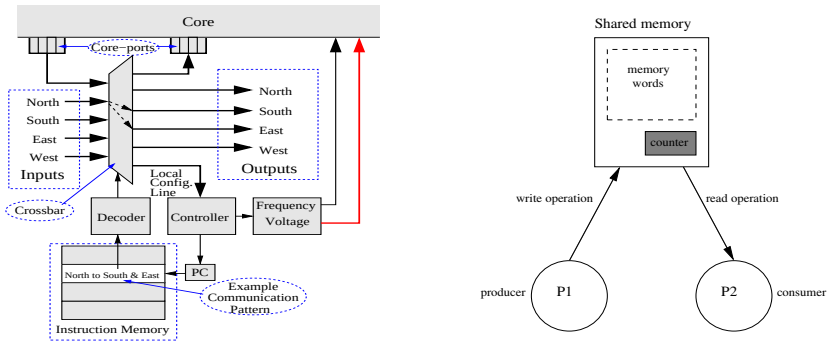
1 Packets with larger payload size will decrease the cache miss rate and consequently decrease the number of packets on the network.

2 Larger packet size will increase the energy consumed per packet, because there are more bits in the payload.

3 Larger packets will occupy the intermediate node switches for a longer time, and cause other packets to be re-routed to non-shortest paths. This leads to more contention that will increase the total number of hops needed for packets traveling from source to destination.

Actually, increasing the cache block size will not decrease the cache miss rate proportionally. Therefore, the decrease of packet count cannot compensate for the increase of energy consumed per packet caused by the increase of packet length. Larger packet size also increases the hop counts on the datapath. Fig. 12.4a shows the combined effects of these factors. The values are normalized to the measurement of 16Byte. As packet size increases, energy consumption on the interconnect network will increase.

The total energy dissipated on the MPSoC is shown in Fig. 12.4b. It clearly decreases as packet size increases. However, when the packets are too large, as in the case of 256Byte in the figure, the total MPSoC energy will increase. This is because when the packet is too large, the increase



(a) ASOC Core Interface

(b) Producer-consumer pair

Figure 12.5. Communication-based power management

of interconnect network energy will outgrow the decrease of energy on cache and memories.

12.6 System and Application Layers

In the context of highly integrated on-chip multi-processors, lowering supply voltage of the cores reduces power quadratically but also results in a performance degradation which can be tolerated only if it does not impact performance beyond a critical, application-dependent threshold. Given the key role played by on-chip communication with respect to MPSoC performance, the concept of *communication-based power management (CBPM)* has been introduced. It consists of integrating the system-level power management functionality into the communication architecture, which binds the system components together, thus eliminating the need for separate power management entities. Second, due to its connectivity, the communication architecture can gather information (such as the execution states of system components) required to make power management decisions. Finally, since the communication architecture schedules inter-component communications, it can control the timing of a component's power modes, thus regulating the component's (and therefore the system's) power profile.

Multiple implementations of this concept are feasible [38, 6], and two relevant examples will be hereafter described. The first one is represented by the Adaptive System-on-Chip (*ASOC*) illustrated in [6], which has been used to build a backbone for power-aware signal processing cores [39]. ASOC ability to provide dynamic voltage and frequency

scaling is due to the architecture of the network interface of the cores, illustrated in Figure 12.5(a) [39]. The core interface uses a synchronized global communication schedule to manage communications through each tile. The instruction memory holds a list of the communication patterns required at run-time. A program counter (PC) fetches these patterns in succession and a decoder converts them into switch settings for a crossbar, that routes data between the local core and neighboring tiles (North, East, South or West). Moreover, at each core, frequency and voltage are automatically adjusted. A subsystem uses up/down counters to track the data transfer rate between core and interconnect. Blocked or unsuccessful transfers cause the count to increase, while successful transfers decrease the value. If the core input port is blocked consecutively, the core is running too slowly with respect to its predecessors. If the core output port is consecutively blocked, the core is running too quickly for its successors. In either case, these counters send trigger signals to the core configuration unit to increase or decrease the core clock. The new frequency setting automatically selects a new supply voltage value.

A similar approach can be applied to pipelined signal processing applications, wherein a sequence of computation stages exchange the results of their processing in a pipelined fashion. From an hardware viewpoint, the system might consist of cascaded producer-consumer pairs communicating by means of a shared memory. If producer and consumer are not well synchronized, energy-inefficient synchronization mechanisms could be triggered. For instance, if the consumer expects input data to process while the producer is not ready to output the result of its computation yet, the consumer keeps polling on a semaphore until its input data is available in the shared memory. This mechanism wastes significant amounts of energy, and should be avoided as much as possible. A solution to keep producer-consumer pairs synchronized is reported in Figure 12.5(b). Shared memory can be abstracted as a queue and a memory access counter keeps track of the queue level. When a lower threshold is crossed, it means that the producer is too slow or the consumer too fast, and frequency/voltage scaling can be applied for balancing data production or consumption rate. The opposite holds when an upper threshold is crossed. Counter monitoring might be carried out by a proper power management hardware module connected to the bus, with the ability to program (through a register) the clock frequency generator of the cores. This could be done continuously or periodically at discrete times, in order to amortize the frequency switching cost. Worst-case power savings with respect to static frequency selection (power-optimized for a particular application) amounts to 12%.

12.7 Application-Specific Networks-on-Chip

Customizing MPSoC architectures and tailoring them to a specific application domain is a very promising approach to system energy minimization. It takes its steps from the optimization techniques used in some SoC design methodologies, that explicitly target applications from a specific domain: their main features (control flow, data organization and type of processing) are evaluated and exploited for a power-aware architecture customization [8].

For customized NoCs to be successful, however, developers must select the appropriate domain-specific architecture and map the system's communication requirements onto it [7]. This is a non-trivial task, in that an optimized network instance has to be derived by analyzing the application communication requirements, and by comparing a number of alternative interconnect solutions.

Moreover, reusing the components of a given NoC architecture across different designs (and therefore network instances) becomes feasible provided the network building blocks (network interfaces, switches, switch-to-switch links) are designed as soft macros.

Some NoC architectures proposed in the literature were built around this concept [6, 4, 3, 2]. Only two of them are mentioned for the sake of brevity. *Quality of Service NoC (QNoC)* [3] is a NoC framework wherein QoS and cost model for communications in SoCs are first defined, and related NoC architecture and design process are then derived. SoC inter-module communication traffic is classified into 4 classes of service: *Signaling* (control signals), *Real-Time*, *RD/WR* (for short data access) and *Block-Transfer* (for large data bursts). By analyzing the communication traffic of the target SoC, QoS requirements (in terms of delay and throughput) for each of the four service classes are derived. A customized QNoC architecture is then created by modifying a generic network architecture (two-dimensional planar mesh, fixed shortest-path multi-class wormhole switching). The customization process minimizes the network cost (in area and power) while maintaining the required QoS and works as follows: the SoC modules are placed so as to minimize spatial traffic density, unnecessary mesh links and switching nodes are removed, and bandwidth is allocated to the remaining links and switches according to their relative load so that link utilization is balanced.

Finally, Xpipes NoC architecture [2] is a library of highly parameterizable network components which are design-time tunable and composable to get customized domain-specific architectures. Xpipes has been designed with high-performance in mind, and this has been achieved by means of deeply pipelined switches, pipelined links to decouple link

throughput from link delay, virtual output buffering. The network interface implements OCP standard signaling and the look-up tables required by static routing algorithms. The network inherently provides best-effort services and targets multi-gigahertz heterogeneous MPSoCs, wherein irregular network topologies with links of uneven length might be required.

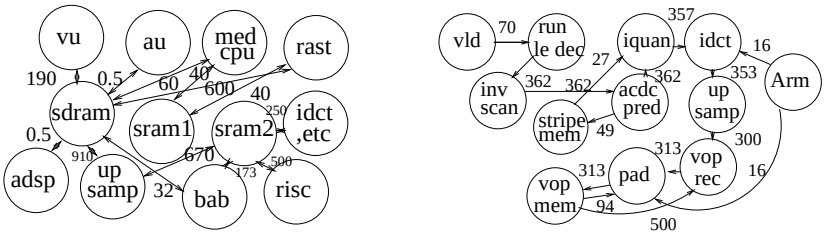
Next, an instructive case study about application-specific NoC instances and their potentials for energy savings is reported, leveraging the Xpipes synthesis flow.

Case study

A core graph representation of the application is the input to Xpipes-based synthesis flow (called `NetChip`). The design and generation of a customized NoC is achieved by means of two tools: `SUNMAP`, which performs the network *topology mapping* and *selection* functions, and `×pipesCompiler`, which performs the *topology generation* function. `SUNMAP` produces a mapping of cores onto various NoC topologies that are defined in a topology library. The mappings are optimized for the chosen design objective (such as minimizing area, power or latency) and satisfy the design constraints (such as area or bandwidth constraints). `SUNMAP` uses floorplanning information early in the mapping process to determine the area-power estimates of a mapping and to produce feasible mappings (satisfying the design constraints). The tool supports various routing functions (dimension ordered, minimum-path, traffic splitting across minimum-paths, traffic splitting across all paths) and chooses the mapping onto the best topology from the library of available ones. A design file describing the chosen topology is input to the `×pipesCompiler`, which automatically generates the SystemC description of the network components (switches, links and network interfaces) and their interconnection with the cores. A custom hand-mapped topology specification can also be accepted by the NoC synthesizer, and the network components with the selected configuration can be generated accordingly.

`NetChip` was applied to two different video processing applications: *Video Object Plane Decoder* (*VOPD* - mapped onto 12 cores), *MPEG4 decoder* (14 cores). These are high-end video-processing applications and the hardware-software partitioning of the applications is presented in [35, 36]. The core graphs of these applications is presented in Figure 12.6. The maximum link bandwidth for the NoCs is conservatively assumed to be 500 MB/s.

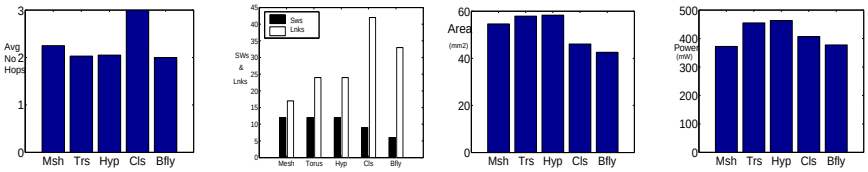
The results of mapping *VOPD* onto various topologies are presented in Figure 12.7. As seen from Figure 12.7(a), the butterfly topology (4-



(a) *MPEG4* core graph

(b) *VOPD* core graph

Figure 12.6. Core Graphs of Video Processing Applications



(a) Avg hop delay

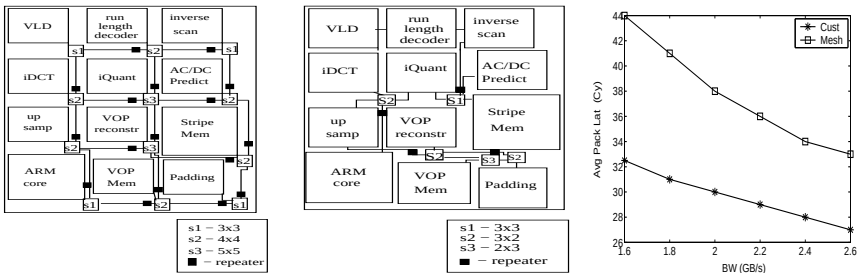
(b) Resource Util

(c) Design Area

(d) Design Power

Figure 12.7. Mapping Characteristics of VOPD

ary 2-fly) has the least communication delay out of all topologies, the least number of switches, but has more links when compared to mesh, torus or hypercube. The large power savings achieved by the butterfly network (Figure 12.7(d)) is attributed to the fact that there are fewer switches and smaller number of hops for communication. Moreover, all the switches are 4x4, while the direct topologies have 5x5 switches. The average link length in the butterfly network (obtained from floorplanner) was observed to be longer than the link lengths (around 1.5×) of direct networks. However, as the link power dissipation is much lower than the switch power dissipation, we get large power savings for the butterfly network. The smaller number of switches and smaller switch sizes also account for the large area savings achieved by the butterfly network. Thus, butterfly is the best topology for *VOPD*. The performance gains for the butterfly over other topologies may be surprising, but careful inspection shows that the butterfly network trades-off path diversity for network switches with average hop delay. On the contrary, the same kind of analysis shows that a mesh topology is more suitable for *MPEG4* than other topologies.



(a) Mesh NoC (b) Appln Specific NoC (c) VOPD Avg Lat

Figure 12.8. Mesh and Custom Mappings of Video Object Plane Decoder

Generating Custom Topologies

For custom topologies, the mapping and generation phases of the tool can be skipped and `×pipesCompiler` can be directly invoked on the input design. A custom hand-tuned NoC for the VOPD is presented in Figure 12.8(b). In the VOPD, about half the cores communicate to more than a single core. This motivates the configuration of this custom NoC, having less than half the number of switches than the mesh NoC. `NetChip` area and power reports relative to the custom NoC were automatically obtained. Significant area ($5x$) and power improvements ($2x$) were noticed with the custom NoC as fewer, smaller size switches are used with respect to the mesh network.

SystemC simulation of the NoC models allowed to assess their performance. The variation of average packet latency (for 64B packets, 32 bit flits and 7 cycle switch delay) with link bandwidth is showed in Figure 12.8(c). Application-specific NoC has lower packet latency as the average number of switch and link traversals is lower. Moreover, the latency increases more rapidly for the mesh NoC with decrease in bandwidth. With the custom NoC, an average of 25% savings in latency (measured at the minimum plotted BW value) is achieved.

12.8 Conclusions

This chapter focuses on low power design techniques for NoC-based gigascale MPSoCs. Several open problems were described at various layers of the communication stack, and the basic strategies to effectively tackle them were sketched. Finally, the large potentials for energy savings provided by the implementation of customized, domain-specific NoCs have been discussed.

References

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

[2] "×`pipes`: a Latency Insensitive Parameterized Network-on-chip A rchitecture For Multi-Processor SoCs", M.Dall'Osso, G.Biccari, L.Giovannini, D.Bertozzi, L.Benini, Int. Conf. on Computer Design, pp.536-541, October 2003.

[3] E.Bolotin, I.Cidon, R.Ginosar, A.Kolodny, "QNoC: QoS architecture and design process for Network on Chip", Journal on Systems Architecture, Special Issue on Networks on Chip, December 2003.

[4] I.Saastamoinen, D.S.Tortosa, J.Nurmi, "Interconnect IP Node for Future System-on-Chip Designs", IEEE Int. Work. on Electronic Design, Test and Applications, pp.116-120, January 2002.

[5] C.T. Hsieh, M. Pedram, "Architectural Energy Optimization by Bus Splitting," *IEEE Trans. CAD*, Vol.21, issue 4, pp.408-414, April 2002.

[6] J.Liang, S.Swaminathan, R.Tessier,"aSOC: A Scalable, Single-Chip Communication Architecture," IEEE Int. Conf. on Parallel Architectures and Compilation Techniques, pp.37-46, October 2000.

[7] S.Murali, G.De Micheli, "Bandwidth-Constrained Mapping of Cores onto NoC Architectures", Design Automation and Testing in Europe, 2004, pp.20896-20901.

[8] L.Bisdounis, C.Dre, S.Blionas, D.Metafas, A.Tatsaki, F.Ieromninon, E.Macii, P.Rouzet, R.Zafalon, L.Benini "Low-Power System-on-Chip Architecture for Wireless LANs," IEE Proc.-Comput. Digit. Tech., Vol.151, no1, January 2004.

[9] K. Lee, S.J. Lee, S.E. Kim, H.M. Choi, D. Kim, S. Kim, M.W. Lee, H.J. Yoo, "A 51mW 1.6GHz On-Chip Network for Low Power Heterogeneous SoC Platform", *IEEE Int.Solid-State Circuits Conference*, pp.1-3, 2004.

[10] S.J. Lee et al., "An 800MHz Star-Connected On-Chip Network for Application to Systems on a Chip ", *IEEE Int.Solid-State Circuits Conference*, pp.468-469, February 2003.

[11] 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.

[12] D. Bertozzi, L. Benini and G. De Micheli, "Low-Power Error-Resilient Encoding for On-chip Data Busses," *DATE, International Conference on Design and Test Europe* Paris, 2000, pp. 102-109.

[13] Dally, W.; Towles, B.; "Route Packets, Not Wires: On-Chip Interconnection Networks" *38th Design Automation Conference, 2001. Proceedings*

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

[15] Dally, W.J; Aoki, H. "Deadlock -free adaptive routing in multicomputer networks using virtual channels" *IEEE Trans. on Parallel and Distributed Systems, April 1993*

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

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

[18] 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.

[19] 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.

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

[21] Karim, F.; Nguyen, A.; Dey, S. "On-chip Communication Architecture for OC-768 Network Processors" *38th Design Automation Conference, 2001. Proceedings*

[22] E. Nilsson "Design and Implementation of a Hot-Potato Switch in a Network on Chip" *Master of Science Thesis, LECS, Royal Institute of Technology*

[23] Li Shang and Li-Shiuan Peh and Niraj K. Jha, "Dynamic Voltage Scaling with Links for Power Optimization of Interconnection Networks," *HPCA - Proceedings of the International Symposium on High Performance Computer Architecture*, Anaheim, February 2003, pp. 91-102.

[24] Singh, J. P.; Weber, W.; Gupta, A,; "SPLASH: Stanford Parallel Applications for Shared-Memory" *Computer Architecture News, vol. 20, no. 1*

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

[26] F. Worm, P. Ienne, P. Thiran and G. De Micheli, " An Adaptive Low-power Transmission Scheme for On-chip Networks," *ISSS, Proceedings of the International Symposium on System Synthesis*, Kyoto, October 2002, pp. 92-100.

[27] 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.

[28] Ye, T. T.; Benini, L.; De Micheli, G.; "Packetized On-Chip Interconnect Communication Analysis for MPSoC" *Design Automation and Test in Europe, DATE 2003 Proceedings*

[29] 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.

[30] 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.

[31] C.H. Zeferino, A.A. Susin, "SoCIN: A Parametric and Scalable Network-on-Chip," *Symposium on Integrated Circuits and Systems Design SBCCI'03*, pp. 169-174, September 2003.

[32] F. Poletti, D. Bertozzi,L. Benini,A. Bogliolo, "Performance Analysis of Arbitration Policies for SoC Communication Architectures," *Journal on Design Automation for Embedded Systems*, Kluwer, pp. 189-210, 2003.

[33] IBM CoreConnect bus architecture, "http://www-3.ibm.com/chips/products/coreconnect"

[34] AMBA Multi-Layer AHB and AHB-Lite, "http://www.arm.com/products/solutions/AMBAAHBandLite.html"

[35] E.B.Van der Tol, E.G.T.Jaspers,"Mapping of MPEG-4 Decoding on a Flexible Architecture Platform", SPIE 2002, pp. 1-13, Jan, 2002.

[36] E.G.T.Jaspers, et al.,"Chip-set for Video Display of Multimedia Information", IEEE Trans. on Consumer Electronics, Vol.4 5, No. 3, pp. 707-716, Aug, 1999.

[37] R.H. Havemann, J.A. Hutchby, "High-Performance Interconnects: An Integration Overview", *Proceedings of the IEEE*, Vol.89, no5, pp.586-601,May 2001

[38] K. Lahiri, A. Raghunathan, S. Dey ,"Communication Architecture Based Power Management for Battery Efficient System Design", Proc. ACM/IEEE DAC, pp.691-696, 2002.

[39] A. Laffely, J. Liang, R. Tessier, W. Burleson ,"Adaptive System on Chip (aSoC): a Backbone for Power-Aware Signal Processing Cores", Int. Conf. on Image Processing, pp.105-108 (III), 2003.

[40] V. Raghunathan, M.B. Srivastava, R.K. Gupta ,"A Survey of Techniques for Energy Efficient On-Chip Communication", DAC 2003, pp.900-905, June 2003.