Networks on Silicon: Blessing or Nightmare?

Paul Wielage and Kees Goossens
Philips Research Laboratories, Eindhoven, The Netherlands
{paul.wielage,kees.goossens}@philips.com

Abstract

Continuing VLSI technology scaling raises several deep submicron (DSM) problems like relatively slow interconnect, power dissipation and distribution, and signal integrity. Those problems are encountered particularly on long wires for global interconnect. As clock frequencies increase, scaled wires become relatively slower, and on-chip communication will be the limiting performance factor of future chips. We explain why efficiently sharing of the wires for long distance communication is the solution to this problem. We introduce networks on silicon (NoS), that route packets over shared (semi)-global wires. NoS performance is expected to be high, but comes at a cost. Balancing the performance and cost of a NoS is a major challenge, and we believe busses still have a role play.

1 Technology trend

VLSI technology scaling has long followed Moore's law. No fundamental barriers have been identified that invalidate this law for at least another decade [12]. Moore's law predicts that chips in 2010 will count over 4 billion transistors, operating in the multi-GHz range. This abundance of transistors will make very complex *systems on silicon* (SoS) possible.

However, challenges at all abstraction levels of design will have to be addressed before such SoSs will become a reality. The three most important deep submicron (DSM) challenges, related to all abstraction levels, are: substantial wire delay, controlling power delivery and dissipation, and assuring signal integrity.

Until recently, on-chip wiring was cheap. Consequently architectural models have been employed that relied on low-latency communication to globally share expensive computational resources. Global wire delay stays at best constant under technology scaling and hence these wires become effectively slower compared to a gate delay. For example, for 130 nm technology the reachable distance of a repeated global signal in a clock cycle is no more than the length of a chip [4]. For 50 nm technology, crossing a chip with highly optimized interconnect takes between six and ten clock-cycles, clearly invalidating the low-latency assumption of today. Hence we must move to system-level architectures that scale with technology.

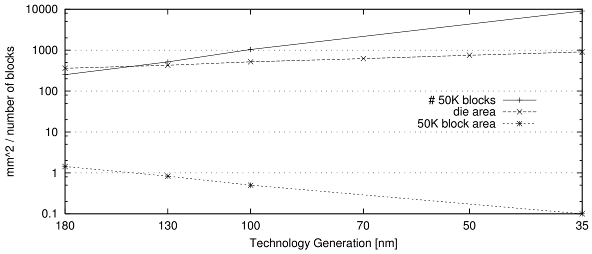

Figure 1. The number of 50k blocks for future process technologies.

A feasible template for a future-proof architecture is constructed from processing nodes that do not grow in complexity with technology. Instead, as technology scales, the number of these processing nodes on the chip grows. An on-chip communication network then combines these nodes into a SoS [4].

Various publications show that the spanning wires in blocks of 50k gates scale with technology [4, 13]. This means that the aforementioned DSM issues can be handled by CAD tools, assuming their evolutionary improvement. Figure 1 shows the exponentially increasing amount of such 50k blocks for a large die in subsequent technologies; in 35 nm this number is approximately ten thousand (adapted from [13] and [4]). It remains to find a communication architecture that allows a SoS composed of these blocks cooperate efficiently.

2 Networks on silicon are inevitable

Given the growing demand for and impact of interconnect on system cost and performance, it is worthwhile to optimize the utilization of wires. Ad-hoc global wiring structures often lead to a huge number of wires with an average usage as low as 10% in time [2]. To control cost in this scenario, the wire packing density must be very high, which is not beneficial for the power and delay characteristics. Efficient mechanisms for sharing (semi)-global wires must solve this cost-performance dilemma.

In deep submicron technologies, (semi)-global wires need special attention for power, signal-integrity, and performance reasons. In the discussion below we show how special circuit techniques can handle these issues. Such techniques only work, however, when embedded in dedicated communication IP, which provides a more abstract interface.

Power is an issue for global interconnect because it costs more energy to send a bit of information over longer the wires. To reduce the communication delay, the energy consumption increases due to bigger drivers. Employing low-swing signaling for the global wires saves up to a factor four in power for these wires [15]. Implementing low-swing signaling requires special circuit techniques.

Signal integrity is hampered increasingly by growing capacitive and inductive coupling between wires. Capacitive noise coupling is the result of the large aspect ratio of wires in DSM technologies. Inductive noise coupling becomes more of a problem due to the decreasing transition times. IR drop[1] in the supply distribution increasingly contributes to the noise. The most effective way to make a connection robust against noise is application of differential signaling [7]. Differential signaling improves both the generation of and sensitivity to noise.

The signal propagation delay of an uninterrupted wire grows quadratically with its length; hence from a certain length onwards it is advantageous to partition the wire in segments with repeaters in between. The repeater insertion technique improves bandwidth and latency but at the cost of higher power consumption. Wire delay can be reduced by fat wires with a lower resistance per unit length at the cost of lower wire density. Such wires behave like lossy transmission lines and require drivers with a resistance matched to the transmission line.

As a result, we believe that all inter-block communication will be implemented by hard-macro transmitters and receivers, employing low-swing differential signaling, with well-controlled interconnect instead of ad-hoc drivers handled by standard place-and-route tools. In this way, communication links can be realized with predictable performance and DSM robustness.

Currently, the prevalent on-chip interconnects are busses [1]. In a bus architecture, devices share a single transmission medium to communicate. At a given time,

[1]Supply voltage drops are caused by high currents (I) flowing through the resistance (R) of the supply network. Since the supply voltage reduces under scaling IR drop worsens.

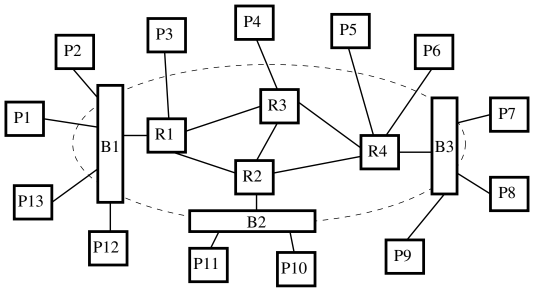

Figure 2. Structural view of a network on silicon consisting of processing nodes (P) and nodes supporting communication (R, B).

only one device has access to the shared medium. An arbitration mechanism is required to order simultaneous accesses. Such functionality is typically performed by a centralized bus arbiter. The performance of a shared-medium bus scales badly. For an increasing number of bus clients (i) individual clients get less bandwidth on average, and (ii) increased capacitive loads and wire length decrease the total bandwidth.

A solution that pairs scalable communication performance and minimal interconnect cost is expected from *networks on silicon* (NoS) where the SoS is considered as a network of components [2, 3, 1]. Figure 2 illustrates the hardware architecture of this concept. The outer components (marked P) exclusively perform processing and storage functions, whereas the inner components (marked B and R) form the NoS and cater to communication needs of the outer components. The basic building blocks of a NoS are routers (R).

A router forwards data from its input ports to its output ports in a concurrent fashion. To that end, a router of arity N contains a $N \times N$ switch matrix. Data packets make their way through the network based on the routing information in their headers. A link between two routers is implemented by a point-to-point connection. The links typically span medium to long distances ranging from several to over more than twenty millimeters. The actual length depends on the chosen topology of the network. For a mesh topology the links are relatively short, for a torus which is a mesh with wrap-around connections, some links have a length of half the edge of the chip. Links can be optimized for bandwidth, latency, power, or a combination of these, depending on performance requirements.

3 NoS requirements

An important characteristic of a future system-level architecture is the separation between computation and communication. A NoS allows the computational blocks to communicate with one other via a uniform interface. A uniform interface is advantageous because (i) it frees the core developer from having to make assumptions about the system in which the core will be used, and (ii) does not constrain the development of newer communication architectures by detailed interfacing requirements of particular legacy SoC components [6]. Several on-chip bus standards are evolving to realize this goal, most notably VCI, put forward by VSIA [14], and more recently, the Open Core Protocol [10].

The fundamental aim of a NoS is to provide flexible and efficient communication between the thousands of IP blocks in a system, with performance guarantees. In a typical SoS, the communication demands of different IP blocks show large variations. For example, data rates may be constant (e.g. digital video) or variable (e.g. compressed video). The importance of latency and jitter also varies greatly. Finally, the data granularity may range from single words to large blocks. A NoS should be able to offer different services to different clients. Each service class must be implemented efficiently, using a shared uniform infrastructure.

A high utilization of the network comes at a price. When the network starts to saturate, throughput and latency will show huge variations, which is not acceptable in real-time applications. Hence, the network should also provide guarantees, like loss-less data transport, minimal bandwidth, and bounded latency. The way packets are buffered and scheduled in routers, and the effects on performance guarantees has been the subject of intense research. Fundamentally, sharing and guarantees are conflicting, and efficiently combining guaranteed traffic with best-effort traffic is hard [11]. Although best-effort services are cheaper than guaranteed services we believe that the latter are essential because they enable compositional and scalable integration of the IP blocks [5]. It is up to the IP integrator at design time, and up to the application at run time, to make a trade off.

4 Performance and cost analysis of NoSs

The vision of previous sections is that the design of future SoSs will allow IP blocks to be plugged in at will to minimize communication costs, but without today's problems like timing closure. In this section we investigate the cost implications of system design based on a NoS. We hope the vision comes at acceptable cost. We hope that the overall cost of a NoS, including the full protocol stack to use it, turn out to be acceptable such that the integration blessings of NoSs do not change into a cost nightmare.

4.1 Performance

The aggregate bandwidth of a router is the product of the bandwidth per port, BW_{port}, the arity of the router (number of ports), N, and a utilization factor, $\alpha \leq 1$ corresponding to the router arbitration scheme.

$$BW_{router} = \alpha \; N \; BW_{port} \tag{1}$$

We discuss each in turn. The bandwidth per port is determined by the bandwidth of the link and the router data path. In short:

$$BW_{port} = B \; \min(BW_{wire}, BW_{route\_data\_path}) \tag{2}$$

where B is the width of the data path. The combined bandwidth of the B wires of a link is a function of the layout characteristics (e.g. total length), chosen signaling technique, and the budgets for power, delay, and area. A first-order expression for the bandwidth of a repeated global wire optimized for power-delay is

$$BW_{wire} = \frac{1}{3 \cdot 2 \cdot \frac{2}{3} \cdot \mathrm{FO4}} \text{ (bits/sec)} \tag{3}$$

where FO4 is the delay of an inverter driving four equally sized inverters [4]. In a 100 nm technology, this yields 5 Gb/s per wire under worst-case environmental conditions. Notice that the bandwidth of repeated global wires scales with technology because such wires allow (wave) pipelining at the segments.

Running the router data path at 5 GHz is not feasible. An aggressive but realistic frequency is 1.25 GHz corresponding the clock frequency of 50k gates blocks [4]. The critical function in the data path is the $N \times N$ switch. For N up to 20 it meets the 1.25 GHz data rate, using N 1-out-of-N multiplexors. The relaxed demand on the wires of the link can be used to reduce power dissipation and area.

The utilization factor, α, reflects the effectiveness of the router to resolve contention on the links. The queuing strategy, the queue sizes, and the schedule algorithm all strongly influence α. Accordingly, many queuing policies and scheduling algorithms have been presented in the literature. For example, $\alpha = 0.59$ for infinite fifo input queues with uniform and independent traffic. (Virtual) output queuing gives $\alpha = 1$ under the same conditions, but at the cost of larger queues and a more complex scheduling algorithm [8]. Static scheduling techniques like (time-division-multiplexed) circuit switching can also improve the utilization factor.

Hence, in 100 nm technology, the bandwidth of a 32 bit router port is approximately 5 GByte/sec.

4.2 Cost

Three main components contribute to the area cost of a router: the switch, the control logic, and the packet queues.

The switch allows N simultaneous connections from the N inputs to the N outputs which results in B arrays of $N \times N$ wires, giving rise to an $O(N^2)$ area cost.

The control logic of a router is made up of the switch-matrix schedule unit and other configuration logic. The delay of a schedule cycle varies greatly per algorithm (for example, for virtual output queuing from $O(1)$ to $O(N^{5/2})$ [9]); it is important for two reasons. First, it determines the lower bound for latency that a flit[2] incurs to traverse the router. Second, it affects the size of the queues. The longer a schedule cycle, the more data arrive, given a fixed bandwidth of a port BW_{port}. This leads to deeper queues, and higher area cost.

The three aforementioned queuing strategies require queues of size $O(N)$ to $O(N^2)$ flits. Scheduling algorithms perform better with deeper queues, with a decreasing return.

Besides routers, a significant amount of area is consumed by so-called *network interfaces* (NI) modules. These modules translate the IP transactions for a given connection to packets that are sent over the network, and vice versa. Packets can be sent once the payload has been completely accepted by the NI. Hence, the buffers must be dimensioned such that, at least a complete packet for every simultaneously active connection can be stored.

The trade off between utilization α and the cost is a complex one, but of importance to the viability of NoSs.

5 The future role of busses

In sections 1 and 2 we have argued that NoSs are essential to solve SoS integration in a scalable fashion. While Section 4.2 raised some general cost issues, we will now more concretely consider the trade off between busses and NoSs. *Will packet-switched NoSs completely replace current busses in future SoSs, or will a hybrid approach emerge?* We believe that shared busses may have a role to play in first-level communication (B in Figure 2) for the following reasons.

First, typical IP blocks underutilize the bandwidth capacity of an individual router port. All router ports offer the same bandwidth that is inherent to the architecture, whereas the bandwidth requirements of IP blocks varies greatly. A shared memory module needs typically much higher (peak) bandwidth than a streaming peripheral device. Single word transfers, variable bit rates, bursty IO, and much lower clock rates for IP blocks than for the NoS further waste bandwidth. This means that the communication needs of a number of IP blocks can be aggregated using a bus before the capacity of a network link is reached.

Second, network interfaces are more expensive (in terms of area) than a bus adaptor. Using a bus as a first-level traffic concentrator, trading bus adaptors for network interfaces thus reduces the overall cost of IP-NoS interfacing. We expect that the overhead of a bus and its network interface are outweighed.

[2] Flit stands for flow control digit, the atomic portion of data handled per schedule cycle. A packet is decomposed in flits.

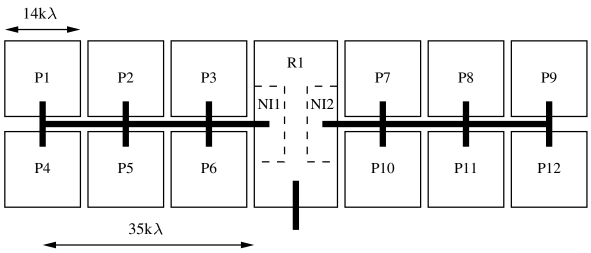

Figure 3. A shared-medium bus seems a cost-effective way to connect the IP to the packet-switched network.

Finally, the number of routers is reduced significantly when busses are used as the first-level interconnect. Routers are larger than busses due to their packet queues and more complex scheduling. We give an example below.

An example of the heterogeneous communication architecture is depicted in Figure 3. A router of arity three surrounded by twelve IP bocks is shown. Two shared-medium busses, each connected to six 50k gates IP blocks, communicate with the router via two network interfaces. These have two functions: first they schedule the transactions on the bus, and second they given the bus clients access to the packet-switched network. The third port of the router provides communication to the remainder of the network. Figure 4 shows an architecture using only routers. Now three routers of arity five and one of arity four are needed.

The suggested shared-medium bus has a length of 35kλ, where λ is half of the length of a minimal transistor. Global wires of this length will not be the bottle-neck of bus performance.[3]

The feasibility of hybrid NoSs hinges on the right implementation of the busses. First, they must be shared wires, as opposed to switches. Second, their arbitration must be combined, or at least compatible with, the scheduling taking place in the network interfaces, to offer uniform end-to-end network services.

We see a future for hybrid NoSs, with first-level communication over a shared-medium bus, and the higher levels using a packet-switched network. Perhaps a packet-switched network can be seen as a distributed and scalable implementation of a logical bridge that connects all the local busses of the SoS. Deciding how many IP blocks can use a local bus

[3] Minimum-delay wire segments have a length of 28kλ, wire segments optimized for power-delay product have a length of 48kλ. These lengths scale with technology as the edge of 50k blocks [4].

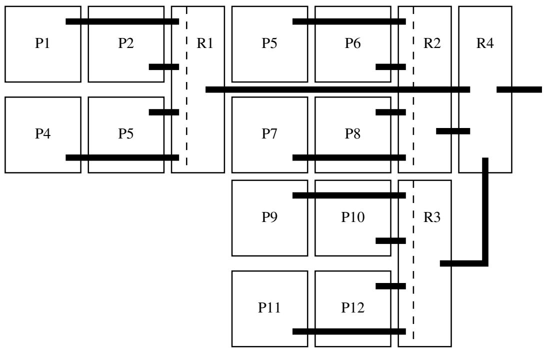

Figure 4. IP to IP communication based on a homogeneous router network.

before connecting to the router network is a question that must be answered foremost.

6 Conclusion

We have argued in Section 1 that future systems on silicon (SoS) will be composed of large numbers of processing nodes (or IP blocks). Each processing node is relatively small (50k gates) to scale with technology, and can be handled by CAD tools, assuming their evolutionary improvement. The interconnect and communication between these blocks then becomes an essential function in itself (Section 2), leading to networks on silicon (NoS). A NoS is based on packet switching to flexibly share link capacity between the network clients, and to provide pluriform communication services over a uniform infrastructure. Both efficiency, provided by best-effort traffic, and predictable performance, such as guaranteed throughput and latency, are important (Section 3). Efficiently combining them is a challenge. Section 4 showed that the performance of a NoS depends on many factors, but is expected to be high. The cost of a NoS can be stated in terms of area (routers, network interfaces), utilization of wires, and speed (latency). They can be traded off against one another, but also, perhaps more interestingly, against the cost of busses. A hybrid NoS using shared-wire busses to communicate locally, and accumulating traffic for a core router network is a promising architecture that deserves to be investigated.

References

[1] L. Benini and G. D. Micheli. Networks on chips: A new SoC paradigm. *IEEE Computer*, 35(1):70–80, Jan. 2002.

[2] W. Dally and B. Towles. Route packets, not wires: On-chip interconnection networks. In *Proc. of DAC*, 2001.

[3] P. Guerrier and A. Greiner. A generic architecture for on-chip packet-switched interconnections. In *Proc. of DATE*, 2000.

[4] R. Ho, K. W. Mai, and M. A. Horowitz. The future of wires. *Proceedings of the IEEE*, 89(4), April 2001.

[5] K. Goossens, J. van Meerbergen, A. Peeters and P. Wielage. Networks on silicon: Combining best-effort and guaranteed services. In *Proc. of DATE*, 2002.

[6] K. Lahiri, A. Raghunathan, and S. Dey. Evaluation of the traffic-performance characteristics of system-on-chip communication architectures. In *Proc. of Int. Conf. on VLSI Design*, 2001.

[7] Y. Massoud, J. Kawa, D. MacMillen, and J. White. Modeling and analysis of differential signaling for minimizing inductive cross-talk. In *Proc. of DAC*, 2001.

[8] N. McKeown, A. Mekkittikul, V. Anantharam, and J. Walrand. Achieving 100% throughput in an input-queued switch. *IEEE Transactions on Communications*, 47(8), August 1999.

[9] N. W. McKeown. *Scheduling Algorithms for Input-Queued Cell Switches*. PhD thesis, University of California, Berkeley, 1995.

[10] Open core protocol specification version 1.0. http://www.sonicsinc.com, 1999.

[11] J. Rexford and K. G. Shin. Support for multiple classes of traffic in multicomputer routers. In *Proceedings of the Parallel Computer Routing and Communication Workshop*, pages 116–130, May 1994. Lecture Notes in Computer Science 853.

[12] S. Rusu. Trends and challenges in vlsi technology scaling towards 100nm (invited paper). In *Proc. of ESSCIRC*, 2001.

[13] D. Sylvester and K. Keutzer. Impact of small process geometries on microarchitectures in systems on a chip. *Proceedings of the IEEE*, 89(4), April 2001.

[14] On chip bus attributes specification 1 OCB 1 1.0, on-chip bus DWG. http://www.vsi.org/library/specs/summary.htm.

[15] H. Zhang, V. George, and J. M. Rabaey. Low-swing on-chip signaling techniques: Effectiveness and robustness. *IEEE Transactions on VLSI*, 8(3), June 2000.