The Growing Cost of Wires

Jonah Probell

Arteris, Inc.[*] Sunnyvale, CA, USA

jonah@arteris.com

ABSTRACT

The cost to floorplan size of wire routing is approaching that of cell area, and therefore must be considered during interconnect design.

The long wires interconnecting IP cores constrain the placement of standard cells and macros. Routing congestion limits the compactness of the chip floorplan. Design automation tool vendors have invested heavily in developing physical synthesis tools that improve locality and decrease average wire lengths. However, the logical design of a network-on-chip interconnect minimizes the number of wires in the first place.

A network-on-chip serializes the transmission of address, control, and data signals over links of configurable width. Trade-offs can be made between throughput and latency. This adds a new dimension for optimizing the design of interconnects.

[*] *The work for this paper was sponsored by Arteris, Inc., a vendor of a tool and libraries for the automated generation of NoC IP and the exploration of the effects of topology on SoC performance.*

1 Introduction

The congestion of wires in the place and route (P&R) stage of chip design poses an increasingly significant challenge to creating low cost, high performance chip designs. High congestion requires an increased die size or more metal mask layers. Congestion causes long wire routes that create new and unpredictable critical paths.

As more IP cores are added to each generation of chips, the number of routes between cores grows by a squared ratio. This increases the problem of congestion. Expectedly, eliminating congestion has become a key concern for EDA tool vendors who have addressed the problem with physical synthesis tools.

The capability to easily configure the widths of interconnection links enables a trade-off between data throughput and wires. For a majority of IP interconnections, this allows a large reduction in the number of long wires at the RTL stage, before synthesis.

This paper presents trends in technology, introduces packet based network on chip as a means of enabling configuration link widths, shows experimental results, and describes other benefits of packet-based interconnect networks.

2 Trends

The highest performance supercomputer, as of this writing, has 18,688 processing nodes [1]. That’s a lot, but it’s still not enough for an accurate weather forecast. Why can’t the operator simply double the number of nodes in order to increase the performance? The answer is: wires. Wires take up space between the processing nodes and disks; sending signals across long wires takes time; driving signals on wires consumes power; and how to connect processing nodes for the best performance is complicated. Solving those problems is the science of supercomputers. Multicore SoCs present analogous problems.

COST

Standard cell sizes are shrinking at a rate of a factor of 2 every 2 years. Wire widths are growing [2]. The P&R inefficiency (I) in die area usage can be calculated from the total die area (A) and the area of the die with wires and no cells (W).

$$I = W / A \tag{1}$$

I is increasing. This is causing a trend in that chip floorplans are larger than they could be. Larger floorplans decrease dies/wafer and increase defects/die.

One way to decrease I is to add more metal layers to the fab process. An increase in layers and their additional masks adds to the manufacturing expense and increases defect rates. In short, wires increase the cost of making chips.

SPEED

The drive strengths of transistors are decreasing faster than wire lengths and die sizes. As a result, signal propagation time along wires is a growing portion of critical path timing. This limits the potential speed improvements of process technology shrinks.

Furthermore, congestion causes some wire routes to take extra long paths. The increasing gap between transistor switching time and signal propagation time along unpredictably routed paths decreases the predictability of the timing of arcs in logic synthesis. Decreased predictability increases the clock skew and the number of hold time violations that must be fixed in P&R.

These timing problems can be reduced with more iteration of synthesis and P&R. However, the eventual tape-out deadline limits those iterations, leaving clock speed on the table.

TOPOLOGY

To add functionality, an increasing the number cores (N) are used in chips. Typically, and with little deviation (anecdotally), about half of the interconnectivity of a full crossbar are used in chip designs. This creates an approximately $N^2/2$ number of logical connections between cores. Using a traditional slave-side arbitrated crossbar, directly connecting every master-slave pair that shares a logical connection would require an intolerable number of wires.

The number of direct connections can be reasonably limited by creating localized crossbars with bridges. This adds overhead to the architectural performance of the chip and can create bottlenecks that limit performance for some use cases. Grouping and bridging requires careful architectural analyses of chip performance, which are significant verification projects.

EDA

Wire routing congestion is the problem of the decade for the vendors of logic synthesis and P&R tools. This is the problem addressed by Cadence Encounter RTL Compiler with Physical and Synopsys Design Compiler Topographical (physical synthesis) / Design Compiler Graphical (floorplan exploration on top of DC-Topo) [3][4]. Consideration of wire routing congestion within EDA tools is essential to minimizing floorplan size and achieving the best possible performance in advanced chip designs [5].

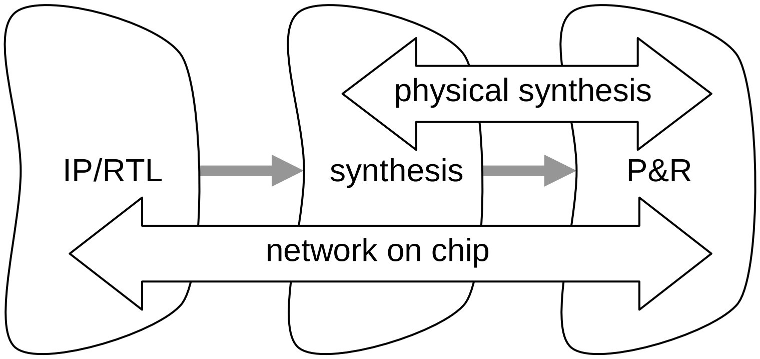
As a general rule for any design process, a small effort earlier in the process yields a better benefit than a large effort later in the process.

> "Upstream tools need to know what's going to happen downstream"
>
> "Congestion is the killer of schedules"
>
> - Aart DeGeus, Synposium 2010

In the process of chip design, more can be done to improve P&R wire congestion by reducing the number of wires in the IP RTL before synthesis.

3 Network on Chip

Reducing P&R congestion from the stage of IP/RTL is the raison d'être of a network on chip (NoC).



The AMBA 4 AXI protocol well serves the needs of all of the many features of modern processors as well as a wide range of other IPs. AXI uses 272 signals for a bidirectional 64-bit interface.

data width	**32**	**64**	**128**
AWID	4	4	4
AWADDR	32	32	32
AWLEN	4	4	4
AWSIZE	3	3	3
AWBURST	2	2	2
AWLOCK	2	2	2
AWCACHE	4	4	4
AWPROT	3	3	3
AWVALID	1	1	1
AWREADY	1	1	1
WID	4	4	4
WDATA	32	64	128
WSTRB	4	8	16
WLAST	1	1	1
WVALID	1	1	1
WREADY	1	1	1
BID	4	4	4
BRESP	2	2	2
BVALID	1	1	1
BREADY	1	1	1
ARID	4	4	4
ARADDR	32	32	32
ARLEN	4	4	4
ARSIZE	3	3	3
ARBURST	2	2	2
ARLOCK	2	2	2
ARCACHE	4	4	4

ARPROT	3	3	3
ARVALID	1	1	1
ARREADY	1	1	1
RID	4	4	4
RDATA	32	64	128
RRESP	2	2	2
RLAST	1	1	1
RVALID	1	1	1
RREADY	1	1	1
Total	**204**	**272**	**408**

A four cycle AXI burst write transaction uses the 56-wire write address bus in only ¼ of cycles. The read address bus is unused while the initiator waits for a read response. Only rarely does an initiator perform a series of read and write requests simultaneously. Even then they do not happen for a sustained number of cycles since few target devices, especially SDRAMs, can sustain that much throughput. In fact, in most systems most AXI interface signals are unused for most of the time.

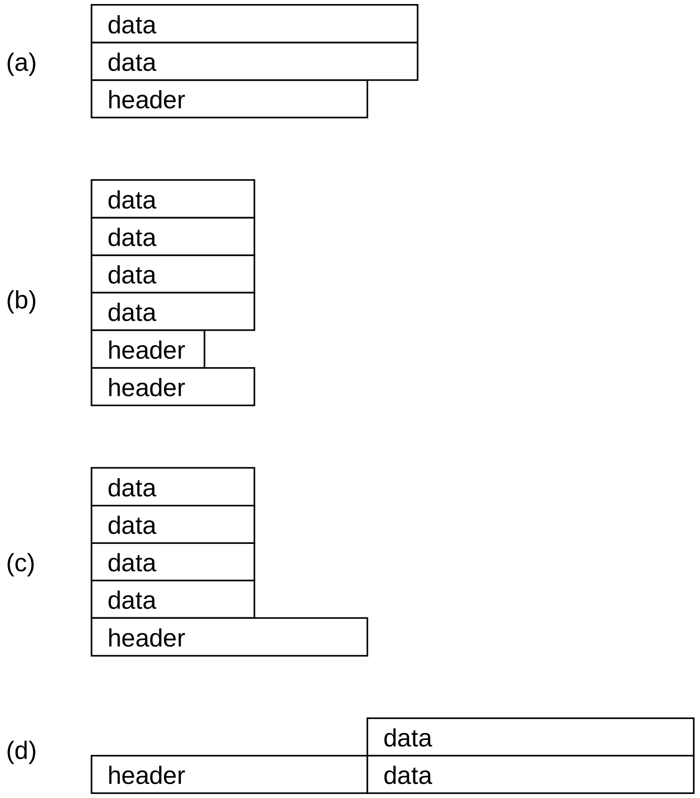
A NoC encodes the address and control signals for a transaction into a header and transports that within a packet that also contains the data. A NoC and its transport protocol are transparent to the initiators and targets within a SoC. In other words, initiators think that they are directly connected to a target and vice versa. A network interface unit (NIU) encodes transactions between the specific IP interfaces using a generic packet transport protocol.

The packet header is usually smaller than the number of address and control signals. This is true because, for example, the upper address bits of a transaction – which determine the target to which the transaction is directed – are encoded in the packet header as a simple route ID. The 18 AXI bits for a burst’s length, size, burst type, lock type, cache type, and protection type can be encoded by the initiator NIU with just several bits needed to signal the packet type to the target NIU.

SERILIZATION

The header and then the data can be transmitted in series on the same wires linking IPs. Thus, serialization decreases wire count on chips.

Serialization offers the opportunity to trade wires for throughput and transaction latency. This gives a degree of freedom not otherwise available to the design of interconnects. In making the trade-offs, four cases of serialization should be considered.



In case (a) the data is wide relative to the header and one cycle of header latency penalty is added to each cycle. An extra cycle of latency is a small consideration in most systems since SDRAM accesses take tens or, if the CPU is fast enough, hundreds of cycles. The throughput penalty of one cycle per transaction is felt in a lightly loaded system, in which case the initiator encounters little traffic to shared targets. In heavily loaded systems, the actual throughput available to each initiator is limited by contention at arbiters, and so the throughput consumed by a header cycle per transaction has no effect.

In case (b) the data width is narrower than the header so that two or more penalty cycles are incurred for each transaction. Such narrow data widths are used on low throughput connections, especially between distant IPs on the die. Since extra cycles will be used for data transport, the extra cycles of header penalty is usually insignificant.

For links in which a high throughput of narrow data is required, multiple cycles of header penalty are intolerable. Case (c) is used for such connections. The link width is chosen as the width of the header to ensure just a single penalty cycle for the header.

For connections between initiators and targets with very high throughput or very low latency requirements, case (d) is appropriate. In this case the header and data are transported in parallel. This uses the largest number of wires and is appropriate between IP interfaces that are placed very on the die, keeping the wire lengths short. Despite having the largest number of wires used in a NoC, case (d) will almost always use fewer wires than the full AXI interface.

Wide datapaths that span long distances are particularly big contributors to congestion because they comprise a large number of signals that must all take approximately the same route. Changing the serialization of data within the NoC topology allows the data to be sent over long distance links with narrower datapaths. AHB and APB interfaces to and from IO devices at the periphery of a chip can be connected to the main CPU block this way. P&R tools can fit the narrower links through hallways between obstructions where it would otherwise be impossible.

Wide busses are also expensive at asynchronous clock domain crossings. Serialization in the faster clock domain reduces the number of signals crossing the boundary. In big prototype boards with multiple FPGAs, serialization at pin boundaries can enable full interconnectivity of all IPs within the chip. This would otherwise be impossible with large scale integration of IPs.

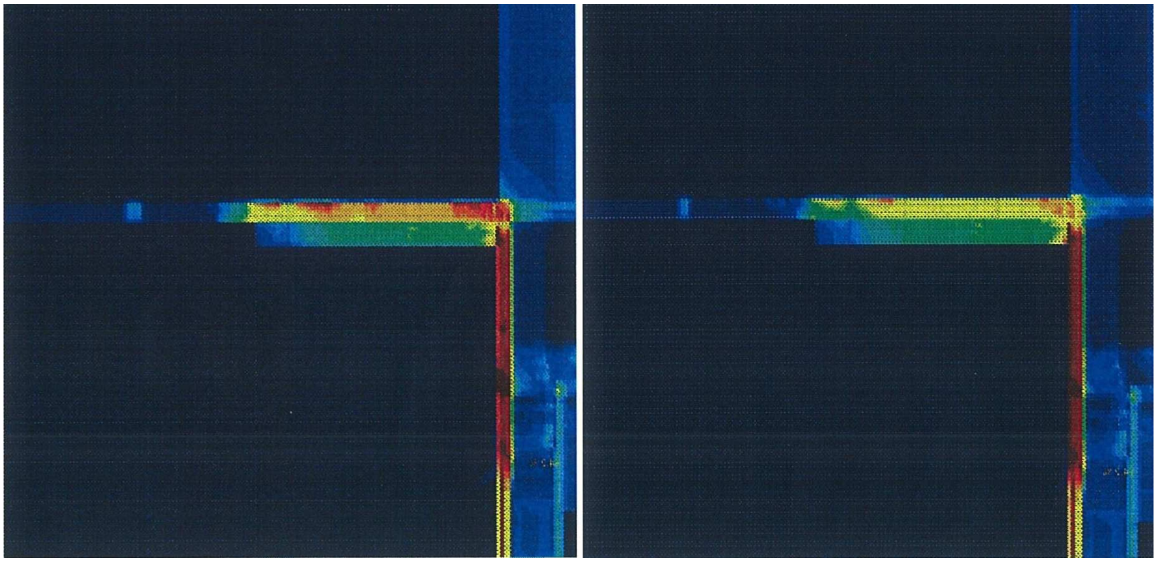
Changing serialization within an interconnect network is possible with traditional interconnects, but the logic to do so is complex. Not only do burst size and length need to be re-encoded, but partial bursts and write enables are hard to handle. Furthermore, single transactions must sometimes be broken into multiple transactions. That has implications for the design of slave devices, which can prevent IP reuse. A NoC benefits from encoding the transaction protocol within a transport protocol.

4 Results

A NoC was built using the Arteris® NoC generator for a chip with 16 initiators and 16 targets, each with bidirectional AXI interfaces. The experiment was conducted for 32-bit, 64-bit, and 128-bit data widths with and without header penalty cycles and without any link serialization changes in the NoC topology. The link widths are shown in the table below.

data width	**32**	**64**	**128**
AXI	204	272	408
Arteris zero header penalty	146	218	362
Arteris auto header penalty	84	156	300

A small chip layout was done with an AXI-based interconnect generator. Twenty design blocks with 64-bit bidirectional AXI interfaces were connected with a total of 5800 interface signals to the AXI-based design and 3000 to the Arteris NoC. A single cycle header latency configuration was used in the NoC. Congestion was measured in the layout. The congestion map in the physical channel between three design blocks is shown below.



Halving the widths of interfaces in the RTL in order to reduce the number of wires allows the EDA tools to reduce peak congestion, shown in red. At the same time, using fewer wires for the same amount of IP logic, allows congestion to be spread more widely.

5 Packetization

Serialization, as discussed above, reduces P&R wire congestion and is the most direct benefit to chip size reduction from packetization. Another benefit of transporting initiator-target transaction as packets is that the arbiter mux logic, which makes up the switch topology within the interconnect network, is simpler and smaller. All of the complexity of describing a transaction and choosing a destination route is done in the initiator NIU. The initiator NIU is placed close to the initiator IP. The complexity of turning transaction packets into the wiggling wires that tickle the target is done in the target NIU. That target NIU is placed close to the target IP. All demuxing and muxing of routed packets throughout the chip is a simple matter of selecting a connection for the packet header and maintaining it until the last cycle of the packet. That uses just a few gates and flip flops of control logic at every switching point in a NoC topology.

Flow control within a NoC is likewise simplified. Packet transmission through a NoC can be controlled by a simple ready-valid handshake at every link point without decoding any other control signals. Mux arbitration within a NoC is no different than arbitration in any other type of interconnect network. Any arbitration scheme can be used with any input prioritization at any point within the NoC topology

Another benefit of the simplicity of the link protocol in a packet-based NoC is the simple insertion of pipeline stages. The busses within the links of a NoC are entirely untyped. They can carry header information in once cycle and data in another. Neither the links nor the switches within the NoC topology know or care about the transport protocol. As a result, pipeline stages can be inserted anywhere in a NoC as a simple register stage.

The NIU logic and all links can be completely combinatorial for low transaction latency in a small chip when low clock speed is required. For high clock speed designs any number of pipeline stages can be inserted within the NoC datapaths. A NoC supports arbitrarily high clock speed. This might include repeaters on long cross-chip wires. The easy insertion of pipeline stages allows another degree of freedom in exploring the trade-offs between clock speed and transaction latency.

6 Conclusion

The advantages of a packet based NoC, particularly serialization, are achieved more easily and to a greater extent than with other approaches to interconnection networks. The easy configuration of the link widths allows a trade-off between throughput and wire congestion. This provides a new degree of freedom in the multidimensional problem of optimizing chip layouts.

The use of a simple transport protocol within a NoC simplifies routing and flow control within the topology. The use of untyped wires simplifies critical path optimization through the insertion, moving, and removal of pipeline stages.

Overall, the use of a packet-based NoC between IP cores produces a more nearly optimal chip design, in terms of cost and performance, with a shorter design time.

7 References

[1] http://www.top500.org/

[2] DAC 2010, Alpert, Charles and Tellez, Gustavo, The Importance of Routing Congestion Analysis

[3] Cadence Encounter RTL Compiler with Physical datasheet
http://www.cadence.com/rl/Resources/datasheets/rtl_physical_ds.pdf

[4] Synopsys Insight Issue 2, 2010, Article 3, Doubling Productivity of of Synthesis and P&R with Design Compiler 2010

[5] Watson, Philip and Fairbairn, Tom; ARM Ltd. and Synopsys respectively, paper DC Graphical: The Promise and The Reality, Synopsys Users Group Boston, 2008.