NoC Advantages for SoC Prototyping on Big FPGA Boards

Jonah Probell
Arteris, Inc
jonah@arteris.com

Abstract

Network-on-Chip interconnect structures use fewer wires than traditional crossbar architectures. A NoC also requires fewer pins to connect links between chips. Big FPGA SoC prototyping boards are constrained by pin resources. SoCs that require many interconnected FPGAs achieve improved interconnect utilization with a NoC architecture. This allows more functionally realistic prototypes of complete SoCs by modeling more IP core interconnections.

I. NoC Architecture

A Network-on-Chip (NoC) uses packets to transfer data between IP core interfaces within a chip. Data is transferred as transactions, requested by initiators and responded to by targets. A NoC can comprise interface adapters, switches, FIFOs, arbiters, multiplexers, rate adapters, bandwidth controllers, memory schedulers, and other component modules. A NoC can have any width of internal interfaces between modules and those physical link bus signals are not typed. That is the fundamental difference between a NoC and other interconnect architectures.

The packets transmitted over the NoC links are made up of a header and, for some packet types, a payload. The header contains address and control signals for routing transactions along paths through the NoC. Typically, the header is transmitted first and the payload follows in successive clock cycles on the same link wires. Like a crossbar, a wide link can send address and control signals in each clock cycle. In that respect, crossbar interconnect architectures are a class of NoC; specifically, the class with the largest number of required wires. NoCs have the benefit of freedom to trade off wires for quality-of-services. QoS are the bandwidth and latency requirements of the system.

In an addendum to this paper, figure 5 shows link widths, cycles, and header latency for 32-byte writes from the request path of a 64-bit AXI initiator. Diagram (a) shows the crossbar approach of fully parallel data, address, and control signaling. It achieves zero header latency and the maximum throughput of 32 bytes per 4 cycles with a link width of 135 signals. Diagram (b) uses nearly half as many wires with a single cycle of header latency and throughput of 32 bytes per 5 cycles. Diagram (c) uses 32 wires with a header latency of 3 cycles and throughput of 32 bytes per 11 cycles.

II. Pin Congestion

To support many different types of designs, modern FPGAs have multiple levels and many routes for signals within the chip. This ensures that the size of designs that can be implemented in the FPGA is usually constrained only by logic element, memory, and I/O pin resources.

The amount of logic and memory resources available for the SoC prototype can be increased by partitioning the design between multiple FPGAs. However, interconnecting FPGAs constrains routing resources more tightly for signals that traverse the FPGA pins. Worse still is that the FPGA pins used for interchip routing are unavailable for system I/Os.

A NoC, designed with narrower links, requires fewer FPGA pins for paths between IP core interfaces than for the interface signals directly.

III. NoC Structure

The throughput, latency, and congestion of transactions moving through a NoC are determined by the topology of module interconnections. Naturally, the choices of link widths and width-converting modules affects throughput. The number and topology of switches, buffers, and repeater pipeline stages affects the ideal latency. The achievable latency and throughput are determined by traffic congestion. Congestion is a product of the topology and the throughput and burstiness of the connected IP cores. The goal of a NoC architecture design is to minimize average latency while meeting bandwidth and latency QoS requirements.

The physical implementation of the structure of the NoC affects its clock speed. Routing paths between modules and IP cores or between modules and other modules of the NoC topology that span long distances require longer clock periods in order to accommodate the signal propagation time. Pipeline stages add one cycle of latency each, but when placed physically midway between modules and IP cores shorten the maximum signal propagation distance. Because NoCs use simple physical link protocols of untyped signals, the insertion of width converters on links between chips minimizes pin requirements. Likewise, the insertion of repeaters directly at the FPGA pin boundaries where FPGAs operate synchronously minimizes clock cycle periods.

IV. FPGA Prototyping Approaches

A NoC architecture can support more interconnections between IP cores than a crossbar because the links between modules in a NoC have fewer signals. Furthermore, the simplicity and untyped nature of NoC links allows the partitioning links between FPGAs to be chosen to trade off pin usage requirements for clock speed.

On big FPGA boards for SoC prototyping there are two approaches to designing the NoC architecture. A cycle accurate prototype exactly models the module topology of the SoC NoC. In a functionally accurate prototype the NoC topology can be modified for the FPGA mode as long as all IP core connections are madel.

In a cycle accurate prototype the number and topological locations of buffers and switches and the number of pipeline stages within paths are the same in the FPGA model as in the SoC. In a functionally accurate prototype the widths of inter-chip links can be narrowed to accommodate more links. Furthermore, pipeline stages can be used directly at the I/O pins to minimize signal propagation time and achieve the best possible FPGA clock rates. These design decisions maximize the utilization of the FPGA pin interconnections to provide faster running prototypes with larger numbers of interconnected IP cores.

V. An Example System

A 32-bit AXI interface has 206 signals and a 64-bit AXI interface has 274 signals [1]. The Dini Group DN2076K10 ASIC prototyping board [2], with 7 FPGAs, was analyzed. The numbers of single-ended signals between each FPGA and between the FPGAs and system I/Os is shown in figure 1.

		FPGA							
		A	B	C	D	E	F	G	I/O
FPGA	A		**234**	76	**232**	76	36	0	384
	B			**276**	78	**234**	76	0	130
	C				0	118	**336**	**234**	16
	D					**314**	74	0	322
	E						**320**	0	0
	F							192	2
	G								36

Figure 1: Table of pin interconnections between FPGAs and system I/Os. Bold numbers represent short paths between adjacent FPGAs.

A crossbar architecture with 64-bit AXI interfaces could use at most 5 of the FPGAs. The only such configuration would be a data flow pipeline with connections BC, CF, FE, ED as shown in Figure 2.

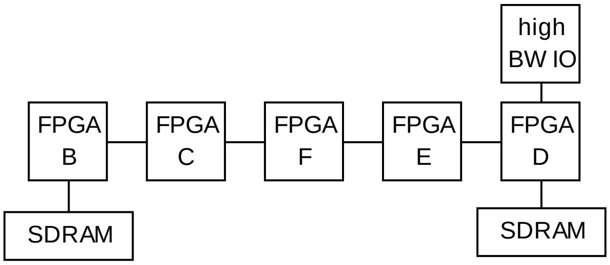

Figure 2: Dataflow pipeline connection of 64-bit AXI interfaces on a DN2076K10 board.

This design is of limited use as it lacks a high-bandwidth input and output flow.

A crossbar architecture design using 32-bit AXI interfaces can model a greater number of realistic interconnections between IP cores in SoCs. Pin constraints would allow initiators to connect to two targets simultaneously for a multi-tree design such as that shown in figure 3.

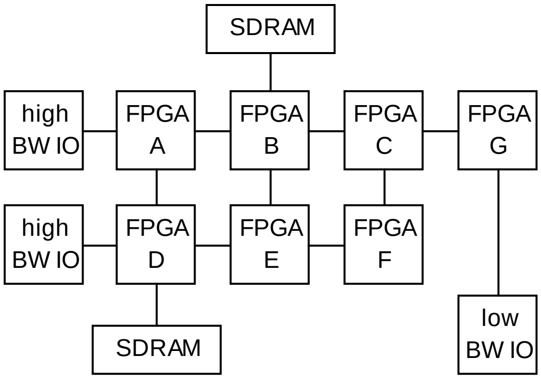

Figure 3: Distributed interonnection of 32-bit AXI interfaces on DN2076K10 board.

The Arteris internal NTTP NoC packet protocol for a 32-bit interface requires 74 signals for each link between IP cores in FPGAs [3]. This allows configurations of IP cores that better model the topology of many SoC designs. As shown in figure 4 below, connections exist between most FPGAs on the board with signaling capacity for 4 IP core links between FPGAs C, F, and E and with 3 link capacity between FPGAs accessing high bandwidth I/Os.

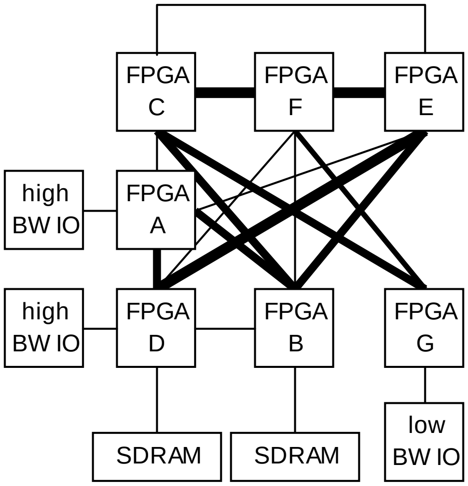

Figure 4: Typical full SoC interconnection of Arteris NoC links possible between FPGAs on DN2076K10 board. Thicker lines represent larger numbers of available links.

Such an interconnection between FPGAs could well accommodate an array of processors in FPGAs C, F, and E. FPGAs A and D could hold high bandwidth system I/O IP cores and DMA controllers. FPGAs B and D could hold 1 or 2 SDRAM controllers with FPGA G holding the low bandwidth interface controllers for the system.

VI. Conclusion

A packet-based NoC architecture provides higher interface signal utilization for the inter-chip connections on big FPGA boards than a traditional crossbar-based interconnect. This enables a greater number of interconnections between IP cores and better inter-chip synchronous clock speeds. As a result, a NoC architecture enables prototyping of larger and more complex SoC designs.

References

[1] ARM; IHI 200C, AMBA AXI Protocol Version: 2.0 Specification, 2010.

[2] The Dini Group; DN2076K10 Block Diagram v1.00, 2010.

[3] Hurt, Shawn; SoC Integration of Heterogeneous IP Using Network-on-Chip, Multicore Expo, 2010.

Addendum

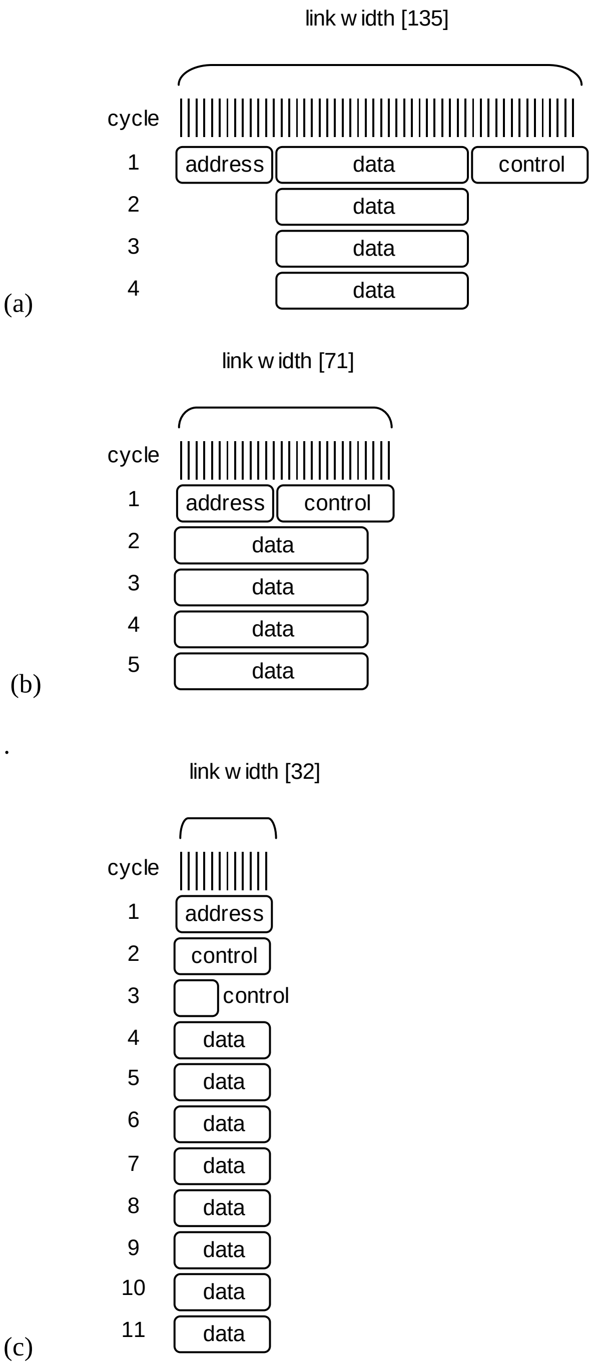

Figure 5: 64-bit AXI initiator write request sequence on 135 (a), 71 (b), and 32 (c) bit links.