Energy-Efficient Embedded NoCs on FPGAs

Mohamed S. Abdelfattah and Vaughn Betz
Department of Electrical and Computer Engineering
University of Toronto, Toronto, ON, Canada
{mohamed,vaughn}@eecg.utoronto.ca

***Abstract*— We propose embedding networks-on-chip (NoCs) on field-programmable gate-arrays (FPGAs) to implement system-level communication. This can especially alleviate the current challenge of connecting the FPGA's fabric to high-speed I/O and memory interfaces, which are a crucial component of FPGA designs. Our mixed and hard embedded NoCs add only ~1% area to large FPGAs and can run much faster than the core logic, thus keeping up with the speed of I/O and memory interfaces. A detailed power analysis, per NoC component, shows that routers consume 14× less power when implemented hard compared to soft, and whether hard or soft most of the router's power is consumed in the input modules for buffering. For complete systems, hard NoCs consume less than 6% (and as low as 3%) of the FPGA's dynamic power budget to support 100 GB/s of communication bandwidth. We find that, depending on design choices, hard NoCs consume 4.5-10.4 mJ of energy per GB of data transferred. Surprisingly, this is comparable to the energy efficiency of the *simplest* traditional interconnect on an FPGA – soft point-to-point links require 4.7 mJ/GB. When comparing a hard NoC against soft buses that are currently used for interconnection We find that for a typical FPGA system, a hard NoC is only at 50% bandwidth utilization yet it still has 4× smaller area and conserves energy compared to soft buses that are currently used for interconnection.**

I. Introduction

Modern FPGAs consist of several million logic cells [1], an assortment of specialized hard blocks such as RAM, multipliers and processor cores, and embedded hard interfaces such as DDRx memory interfaces and PCIe transceivers. These capabilities make FPGAs a strong programmable platform for implementing large complex systems for computation; however, it is still difficult to complete FPGA designs. One of the main difficulties in designing for FPGAs is creating the system-level interconnect; currently this interconnect consists of multiplexer-based soft buses constructed out of the FPGA fabric. It is challenging to create these soft buses that must often connect a hard interface running up to 10 times faster than the FPGA fabric. Because the soft bus is much slower than these interfaces, it must also be very wide to transport the incoming data bandwidth. For example, a single 64-bit DDR3 933 MHz interface requires both a 576-bit wide input and a 576-bit output bus running at over 200 MHz, and these buses often span much of the chip. To design that very wide bus for 200 MHz is a challenge that often necessitates multiple design iterations. Additionally, these huge buses rapidly consume a large fraction of the FPGA's resources; both area and power.

We propose augmenting the FPGA's conventional interconnect with a high-speed embedded network-on-chip (NoC) for the purpose of handling global communication between I/O interfaces, hard blocks and the FPGA fabric (Fig. 1). The NoC abstraction can simplify design and speed up compilation [2, 3]. Our recent work showed that hard NoCs have compelling area and delay advantages over soft NoCs [2]; however, power is a major concern: Does this higher level of interconnect abstraction come at an unacceptable power cost? And how do NoCs compare to the multiplexer-based soft buses that are currently used for interconnection? In answering these questions, we investigate both how to design an energy-efficient NoC in the FPGA context and how the power of this NoC compares to that of the conventional fabric.

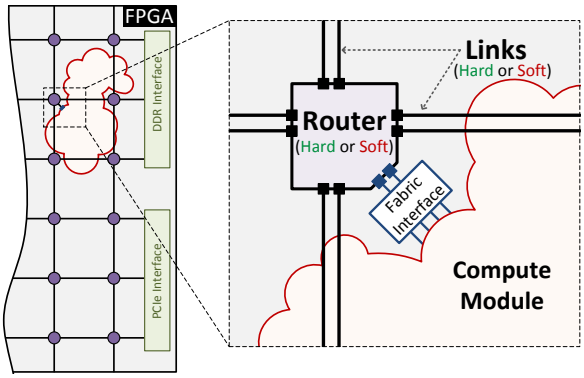


Fig. 1: A mesh NoC implemented on an FPGA. The example shows one router connected to a compute module and three links connected to each of the DDR and PCIe interfaces.

Both soft NoCs [4–6] and hard NoCs [7, 8] have been introduced in the context of FPGAs, but power consumption was seldom analyzed. However, there is an extensive body of work discussing the power consumption of NoCs for multiprocessors. Some papers discuss the power breakdown of NoCs by router components and links, and investigate how power varies with different data injection rates in an NoC [9–11]. Other work focuses on complete systems and reports the power budgeted for communication using an NoC [12, 13]. Finally, NoCs have been compared to other interconnect types by using application-independent metrics, such as the amount of energy to move a unit of data over different kinds of interconnect [14]. We build on some of the concepts introduced in this literature; however, we also address many FPGA-specific questions that were not addressed in any prior work.

After presenting two novel NoC architectures for FPGAs, we perform an in-depth power analysis for both hard and soft NoC components, and how each component's power consumption varies with different design parameters. We then

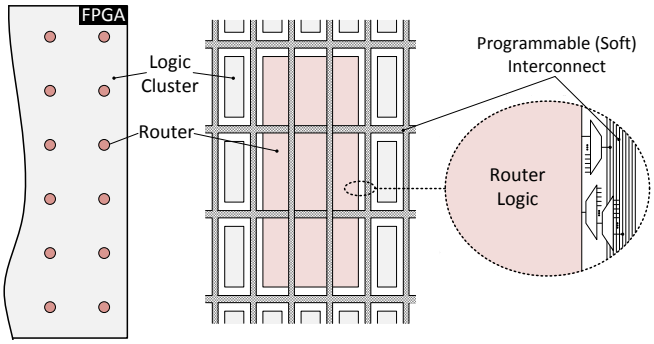


Fig. 2: Floor plan of a hard router with soft links embedded in the FPGA fabric. Drawn to a realistic scale.

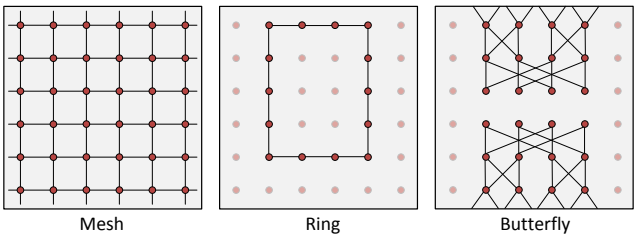


Fig. 3: Examples of different topologies that can be implemented using the soft links in a mixed NoC.

look at power-aware design of complete NoCs and report their power usage as a fraction of the available FPGA power budget. We also investigate how utilization and data congestion of the NoC impacts power consumption, and how the "raw" energy efficiency of NoCs compare to soft point-to-point links on FPGAs. Finally, we compare our hard NoCs to soft buses which are currently used to interconnect FPGA systems, and show that both area and energy can be significantly lowered if a hard NoC is embedded on an FPGA and used for interconnection. Our contributions include[1]:

- Two novel NoC architectures for FPGAs. One uses soft links between routers and the other uses hard links.
- Power analysis of hard and soft NoC components with different design parameters and data rates.
- Design space exploration of power-efficient hard NoCs, taking into account the FPGA's power budget.
- Comparison of NoC energy consumption to regular soft point-to-point links on FPGAs.
- Area and energy comparison between hard NoCs and soft buses that are currently used for system-level interconnection.

II. Network Architecture

NoCs consist of routers and links. Routers perform distributed buffering, arbitration and switching to decide how data moves across a chip, and links are the physical wires that carry data between routers.

[1]An earlier version of this work appeared in [15] but focused only on NoC component analysis. We extend this work here by adding a crucial comparison between hard NoCs and soft buses which are currently used for system-level interconnection. We analyze area, power and frequency trends with different sizes of buses, as well as investigate the overhead of soft buses in a complete representative FPGA system and how that compares to hard NoCs.

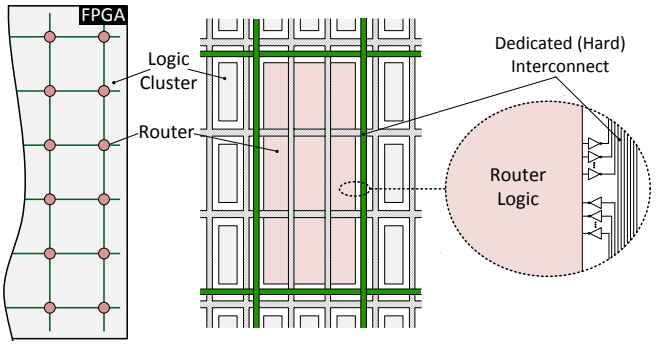


Fig. 4: Floor plan of a hard router with hard links embedded in the FPGA fabric. Drawn to a realistic scale.

On FPGAs, communication bandwidth demands are high. In particular, FPGAs interface to many high-speed I/Os such as DDRx, PCIe, Gigabit Ethernet and serial transceivers. To keep up with these high-throughput data streams and move data across the FPGA with low latency, we base our NoCs on a high-performance packet switched router [16]. This packet-switched router includes a superset of the components that are used in building any NoC. Because we analyze each subcomponent separately, studying this full-featured router yields a more complete analysis of the design space. For details of the router microarchitecture, please see [2, 16].

We investigate the design of NoCs on FPGAs; as shown in Fig. 1 both routers and links can be either soft or hard. Soft implementation means configuring the NoC out of the conventional FPGA fabric while hard implementation refers to embedding the NoC as unchangeable logic on the FPGA chip. We compare the power of soft NoCs to that of several possible hard NoCs. Note that a 64-node version of a hard NoC adds less than 1% area to a large FPGA, making it a highly practical addition [2].

A. Mixed NoCs: Hard Routers and Soft Links

In this NoC architecture, we embed hard routers on the FPGA and connect them via the soft FPGA interconnect. Similarly to logic clusters or block RAMs on the FPGA, a hard router requires programmable multiplexers on each of its inputs and outputs to connect to the soft interconnect in a flexible way. We connect the router to the interconnect fabric with the same multiplexer flexibility as a logic block and we ensure that enough programmable interconnect wires intersect its layout to feed all of the inputs and outputs. Fig. 2 shows a detailed illustration of such an embedded router. After accounting for these programmable multiplexers, mixed NoCs are on average 20× smaller and 5× faster than a soft NoC [2]. Note that the speed of such an NoC is limited by the soft interconnect.

While this NoC achieves a major increase in area-efficiency and performance versus a soft NoC, it remains highly configurable by virtue of the soft links. The soft interconnect can connect the routers together in any network topology. That includes implementing topologies that use only a subset of the available routers or implementing two separate NoCs as

shown in Fig. 3. To accommodate for different NoCs, routing tables inside the router control units are simply reprogrammed to match the new topology.

B. Hard NoCs: Hard Routers and Hard Links

This NoC architecture involves hardening both the routers and the links. Routers are connected to other routers using dedicated hard links; however, routers still interface to the FPGA through programmable multiplexers connected to the soft interconnect. When using hard links, the NoC topology is no longer configurable. However, the hard links save area (as they require no multiplexers) and can run at higher speeds than soft links, allowing the NoC to achieve the router's maximum frequency. Drivers at the ends of dedicated wires charge and discharge data bits onto the hard links as shown in Fig. 4. After accounting for these wire drivers, and the programmable multiplexers needed at the router-to-FPGA-fabric ports, this NoC is on average 23× smaller and 6× faster than a soft NoC. Its speed (above 900 MHz) is beyond that of the programmable clock networks on most FPGAs, accordingly it also requires a dedicated clock network to be added to the FPGA. Such a clock network is fast and very cheap in terms of metal usage since it is not configurable and has only as many endpoints as the number of routers in an NoC; typically less than 64 nodes. In contrast, FPGAs have more than 16 configurable clock networks with ~600 endpoints each.

A hard NoC is almost completely disjoint from the FPGA fabric, only connecting through router-to-fabric ports. This makes it easy to use a separate power grid for the NoC with a lower voltage than the nominal FPGA voltage. This is desirable because we can trade excess NoC speed for power efficiency. The only added overhead is the area of the voltage crossing circuitry at the router-to-fabric interfaces, and this is minimal. In our analysis we explore this hard NoC architecture both at the FPGA's nominal voltage (1.1 V) and, for lower power, at 0.9 V.

III. Methodology

NoC power is consumed in routers and links. We measure the power consumed by those two components both when implemented soft in the FPGA fabric or hard in ASIC gates. The NoC is implemented both on the largest Stratix III FPGA (EP3SL340) and TSMC's 65 nm ASIC process technology. This allows a direct comparison since Stratix III devices are manufactured in the same 65 nm TSMC process [17]. We start with an NoC with the baseline router parameters listed in Table I. We then vary each of the parameters independently to understand how each NoC parameter impacts dynamic power consumption. Note that we only investigate dynamic power and not static power because of the lack of a method to compare static power fairly. Static power dissipation, or leakage, can be arbitrarily controlled by changing the threshold voltage of the transistors, which also affects transistor speed. For this reason, previous work has shown that comparing static power consumption on FPGAs and ASICs draws no useful conclusions [18].

TABLE I: Baseline router parameters.

Width	Num. of Ports	Num. of VCs	Buffer Depth
32	5	2	10 (5/VC)

A. Router Power

We generate the post-layout gate-level netlist from the FPGA CAD tools (Altera Quartus II v11.1) and the post-synthesis gate-level netlist from the ASIC CAD tools (Synopsys Design Compiler vF-2011.09-SP4) as outlined in prior work [2]. For accurate dynamic power estimation, we first simulate these gate-level netlists with a testbench to extract realistic toggle rates for each synthesized block in the netlists.

The testbench consists of data packet generators connected to all router inputs and flit sinks at each router output. The packet generator understands back pressure signals from the router, so it stops sending flits if the input buffer is full. We attempt to inject random flits every cycle into all inputs and we accept flits every cycle from outputs to maximize data contention in the router, thus modeling an upper bound of router power operating under worst-case synthetic traffic. We perform a timing simulation of the router in Modelsim for 10000 cycles and record the resulting signal switching activity in a value change dump (VCD) file. Note that we disregard the first and last 200 cycles in the testbench so that we are only recording the toggle rates for the router at steady state and excluding the warm-up and cool-down periods.

This simulation is very accurate for two main reasons. First, by simulating the gate-level netlist we obtain an individual toggle rate for each implemented circuit block. Second, we perform a timing simulation that takes all the delays of logic and interconnect into account; consequently the toggle rates are highly accurate and include realistic glitching. It is then a simple task for power analysis tools to measure the power of each synthesized block (LUTs, interconnect multiplexers or standard cells) by using their power-aware libraries and the simulated toggle rates on each block input and output.

We use the extracted toggle rates to simulate dynamic power consumption, per router component, for both the FPGA and ASIC using their respective design tools: Altera's PowerPlay Power Analyzer for the FPGA and Synopsys Power Compiler for the ASIC. The nominal supply voltage for the TSMC 65 nm technology library is 0.9 V compared to 1.1 V for the Stratix III FPGA. For that reason, we scale the ASIC dynamic power quadratically (by multiplying by $\frac{1.1^2}{0.9^2}$) when computing FPGA-to-ASIC power ratios. In all other power results, we explicitly state which voltage we are using.

B. Links Power

1) Soft (FPGA) Links: Soft NoC links are implemented using the prefabricated FPGA "soft" interconnect. On Stratix III FPGAs, there are four wire types: vertical length four (C4) and length 12 (C12), and horizontal length four (R4) and length 20 (R20). We connect two registers using a single

wire segment to measure the delay and dynamic power of this wire segment. Next, we investigate different connection lengths by connecting wire segments of the same type in series and measuring delay and power. Registers are manually placed using location constraints to define the wire endpoints, and the connection between the registers is manually routed by specifying exactly which wires are used in a routing constraints file (RCF).

Wire delay is measured using the most pessimistic (slow, 85 ^{o}C) timing model. The dynamic power consumed by the wires is linearly proportional to the toggle rate. 0% means that the wire has a constant value, while 100% means data toggles on each positive clock edge. For each simulated router instance, we extract the toggle rates at its inputs and outputs and use that to simulate the wire power. This ensures that the data toggle rates on the NoC links correctly match the router inputs and outputs to which the links are connected.

2) Hard (ASIC) Links: We use TSMC's metal properties to simulate lumped element models of wires allowing us to measure the delay and power of ASIC NoC links. Metal resistance and capacitance are provided with TSMC's 65 nm technology library for each possible wire width and spacing on each metal layer. Metal layers are divided into three groups based on the metal thickness: local, intermediate and global. In our measurements, we use the intermediate wires because, unlike the alternatives, they are both abundant and reasonably fast. We use Synopsys HSPICE vF-2011.09.SP1 to simulate a lumped element (π) model of hard wires [19]. Propagation delay is measured for both rising and falling edges of a square pulse signal, and the worst case is taken to represent the speed of this wire. Dynamic power is computed using the equation ($P = \frac{1}{T}\int_0^T V\, I(t)\,\mathrm{d}t$) and it is scaled linearly to the routers' toggle rates.

We design and optimize the ASIC interconnect wires to reach reasonably low delay and power comparable to FPGA wires by choosing:

1) Wire width and spacing: Controls the parasitic capacitance and resistance in a wire segment which determines its delay and power dissipation.
2) Drive strength: The channel width of transistors used in the interconnect driver. Affects speed and power.
3) Rebuffering: How often drivers are placed on a long wire.

Using the π wire model, we conducted a series of experiments using HSPICE to optimize our ASIC wire design. To match the FPGA experiments, the supply voltage was set to 1.1 V and the simulation temperature at 85 ^{o}C. We also repeated our analysis at 0.9 V for the low-power version of our hard NoC. We reached a reasonable design point with metal width and spacing of $0.6\ \mu m$, drive strength of 20-80$\times$ that of a minimum-width transistor (depending on total wire length) and rebuffering every 3 mm. If necessary, faster or lower power ASIC wires could be designed with further optimization or by using low-swing signaling techniques [20].

TABLE II: Summary of FPGA/ASIC power ratios.

Module	Min.	Max.	Geometric Mean
Input Module	3	23	10
Crossbar	15	194	64
Allocators	33	61	41
Output Module	14	19	16
Router	5	27	14

IV. Power Analysis of NoC Components

This section investigates the dynamic power of both hard and soft NoC components; only by understanding where power goes in various NoCs can we optimize it.[2] We divide the NoC into routers and links, and further divide the routers into four subcomponents. After sweeping four key design parameters (width, number of ports, number of virtual channels (VC) and buffer depth) we find the soft:hard power ratios for each router component as shown in Fig. 5. We also investigate the percentage of power that is dissipated in each router component for both hard and soft implementations in Figures 6 and 7. Finally, we analyze the speed and power of NoC links (Fig. 9) whether they are constructed out of the FPGA's soft interconnect or dedicated hard (ASIC) wires.

A. Router Power Analysis

1) Router Dynamic Power Ratios: As Table II shows, routers consume 14$\times$ less power when implemented hard compared to soft. When looking at the router components, the smallest power gap is 10$\times$ for input modules since they are implemented using efficient BRAMs on FPGAs. On the other hand, crossbars have the highest power gap (64$\times$) between hard and soft. Note that there is a strong correlation between the FPGA:ASIC power ratios presented here and the previously published NoC area ratios, while the power and delay ratios do not correlate well [2]. We believe this is because total area is a reasonable proxy for total capacitance, and charging and discharging capacitance is the dominant source of dynamic power.

Width: Fig. 5 shows how the power gap between hard and soft routers varies with NoC parameters. The first plot shows that increasing the router's flit width reduces the gap. For example, 16 bit soft crossbars consume 65$\times$ more power than hard crossbars, while that gap drops to approximately 40$\times$ at widths higher than 64 bits. The same is true for input modules where the power gap drops from 18-12$\times$. This indicates that the FPGA fabric is efficient in implementing wide components and encourages increasing flit width as a means to increase router bandwidth when implementing soft NoCs.

Number of Ports: Unlike width, increasing the number of router ports proved unfavorable for a soft router implementation. The allocators power gap is 57$\times$ at high port count compared to 35$\times$ at low port count. For crossbars, the power

[2]To access and visualize our complete area/delay/power results, please visit: www.eecg.utoronto.ca/~mohamed/noc_designer.html

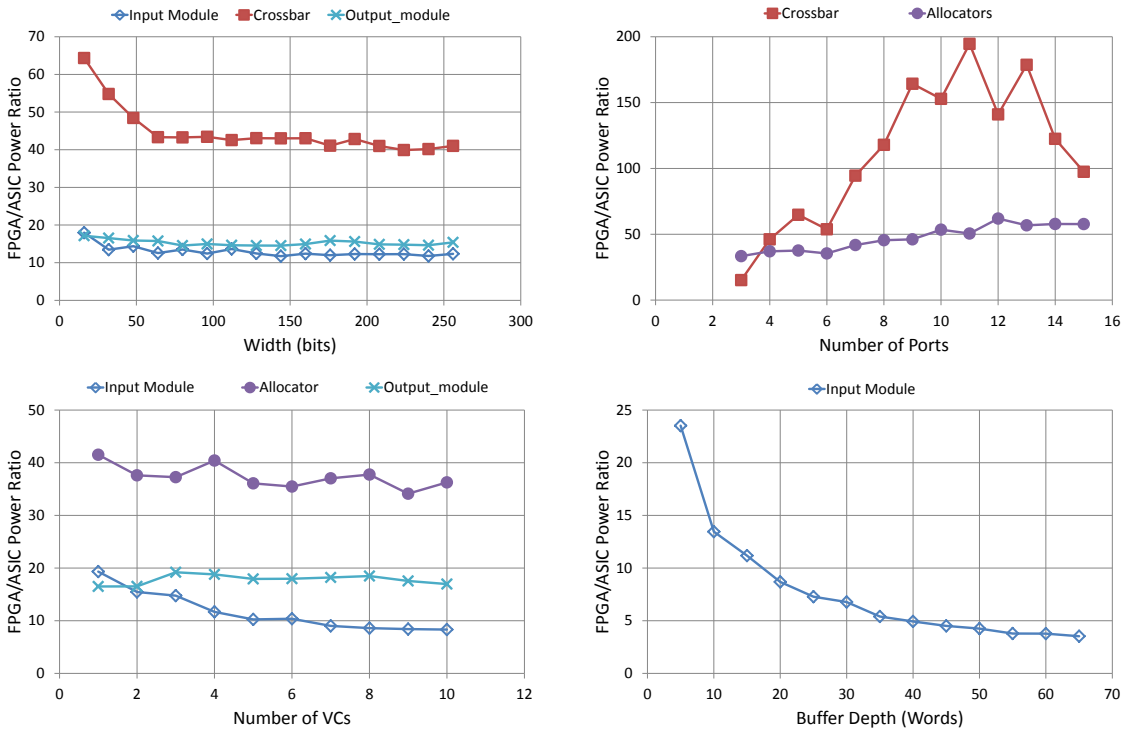


Fig. 5: FPGA/ASIC (soft/hard) power ratios as a function of key router parameters.

gap triples from 50× at six or less ports, to 150× with a higher number of ports. This suggests that low-radix soft NoC topologies, such as rings or meshes, are more efficient on traditional FPGAs than high-radix and concentrated topologies.

Number of VCs and Buffer Depth: Increasing the number of VCs is another means to enhance router bandwidth because VCs reduce head-of-line blocking [21]. This requires multiple virtual FIFOs in the input buffers and more complex control and allocation logic. Because we use BRAMs for the input module buffers on FPGAs, we have enough buffer depth to support multiple large VCs. Conversely, ASIC buffers are built out of registers and multiplexers and are tailored to fit the required buffer size exactly. As a result, the input module power gap consistently becomes smaller as we increase the use of buffers by increasing either VC count or buffer depth, as shown in Fig. 5.

Allocators are composed of arbiters, which are entirely composed of logic gates and registers. Increasing the number of VCs increases both the number of arbiters and the width of each arbiter. The overall impact is a weak trend – the power ratio between soft and hard allocators narrows slightly as the number of virtual channels increases.

2) Router Power Composition: Figures 6 and 7 show the percentage of dynamic power consumed by each of the router components and the total router power is annotated on the top axes. Clearly most of the power is consumed by the input modules, as shown by previous work [9, 14], but the effect is weaker in soft NoCs than in hard. This also conforms with the area composition of the routers; most of the router area is dedicated to buffering in the input modules, while the smallest router component is the crossbar [2]. Indeed, the crossbar power is very small compared to other router components as shown in the figures.

Next we look at the power consumption trends when varying the four router parameters. As we increase width, the router datapath consumes more power while the allocator's power remains constant. When increasing the number of ports or VCs, the proportion of power consumed by the allocators increases since there are more ports and VCs to arbitrate between. With deeper buffers, there is almost no change in the soft router's total power or its power composition. This follows from the fact that the same FPGA BRAM used to implement a 5-word deep buffer is used for a 65-word deep buffer. However, on ASICs there is a steady increase of total power with buffer depth because deeper buffers require building new flip-flops and larger address decoders.

3) Router Power as a Function of Data Injection Rate: Router power is not simply a function of area, it also depends very strongly on the amount of data traversing the router. A logical concern is that NoCs may dissipate more energy per unit of data under higher traffic. This stems from the fact that NoCs need to perform more (potentially power consuming) arbitration at higher contention levels, with no increase in data packets getting through. However, our measurements refute that belief. Fig. 8 shows that router power is linear with the amount of data actually traversing the router, suggesting that higher congestion does not raise arbitration power. We annotate the attempted data injection rate on the plot. For example, 100% means that we attempt to inject data on all router ports on each cycle, but the x-axis shows that only

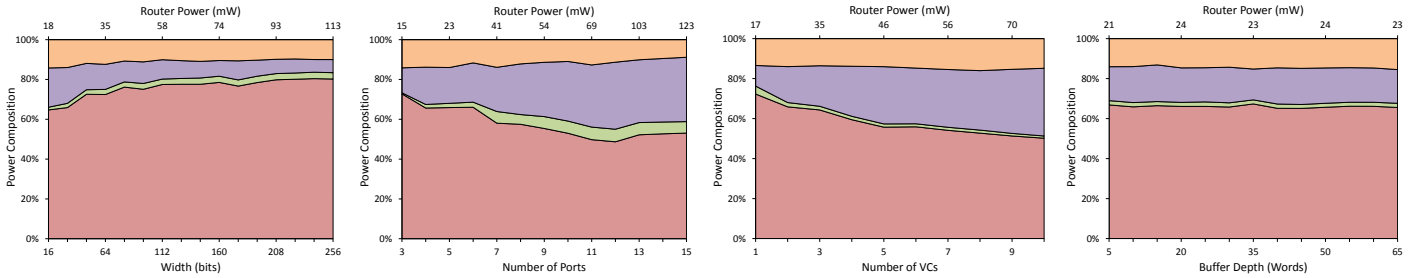


Fig. 6: FPGA (soft) router power composition by component and total router power at 50 MHz. Starting from the bottom (red): Input modules, crossbar, allocators and output modules.

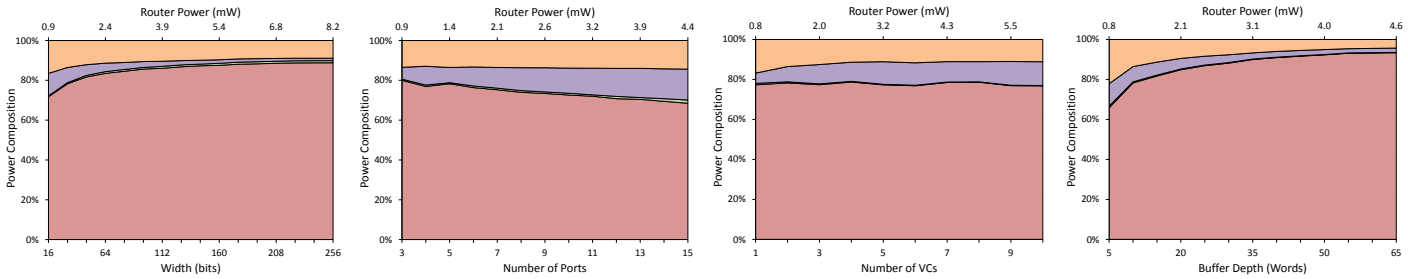


Fig. 7: ASIC (hard) router power composition by component and total router power at 50 MHz. Starting from the bottom (red): Input modules, crossbar (very small), allocators and output modules.

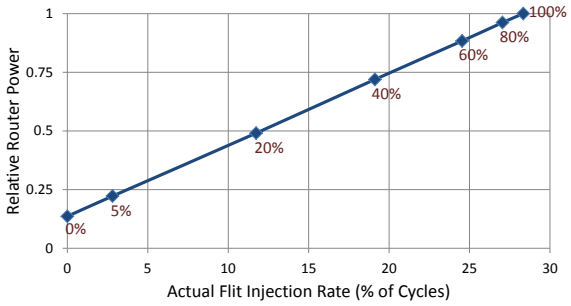


Fig. 8: Baseline router power at actual data injection rates relative to the its power at maximum data injection. Attempted data injection is annotated on the plot.

28% of the cycles carry new data into the router. At zero data injection the router standby power, because of the clock toggling, is 13% of the power at maximum data injection, suggesting that clock gating the routers is a useful power optimization [10]. Importantly, router parameters also affect the data injection rate at each port.

- ***Width:*** Increasing port width does not affect the data injection rate because switch contention does not change. However, bandwidth increases linearly with width.
- ***Number of ports:*** Increasing the number of ports raises switch contention; thus the data injection rate at each port drops from 38% at 3 ports to 19% at 15 ports.
- ***Number of VCs:*** At 1 VC, data can be injected in 22% of the cycles and that increases to 32% at 4 VCs. Beyond 4 VCs, throughput saturates but multiple VCs can be used for assigning packet priorities and implementing quality of service guarantees [21].
- ***Buffer Depth:*** While deeper buffers increase the number of packets at each router, it does not affect the steady-state switch contention or the rate of data injection.

B. Links Power Analysis

Fig. 9 shows the speed and power of hard and soft wires. Soft wires connect to multiplexers which increases their capacitive and resistive loading, making them slower and more power hungry. However, these multiplexers allow the soft interconnect to create different topologies between routers, and enables the reuse of the metal resources by other FPGA logic when unused by the NoC. We lose this reconfigurability with hard wires but they are, on average, 2.4× faster and consume 1.4× less power than soft wires. We can also trade excess speed for power efficiency by using lower-voltage wires as seen from the "Hard 0.9V" plots.

A detailed look at the different soft wires shows that long wires (C12, R20) are faster, per mm, than short wires (C4, R4). Additionally there is a directional bias for power as the horizontal wires (R4, R20) consume more power per mm than vertical ones (C4, C12). An important metric is the distance that we can traverse between routers while maintaining the maximum possible NoC frequency. This determines how far we can space out NoC routers without compromising speed. In the case of soft links and a soft (programmable) clock network, the clock frequency on Stratix III is limited to 730 MHz. At this frequency, short wires can cross 3 mm while longer wires can traverse 6 mm of chip length between routers. When using hard links, we are only limited by the routers' maximum frequency, which is approximately 900 MHz. At this frequency, hard links can traverse 9 mm at 1.1 V or 7 mm at 0.9 V. Although lower-voltage wires are slower, they conserve 40% dynamic power compared to wires running at the nominal FPGA voltage.

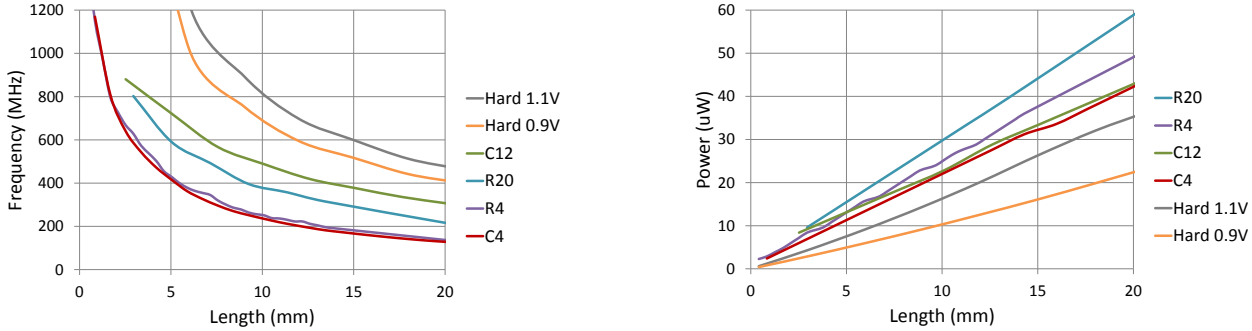


Fig. 9: Hard and soft interconnect wires frequency, and power at 50 MHz and 15% toggle rate.

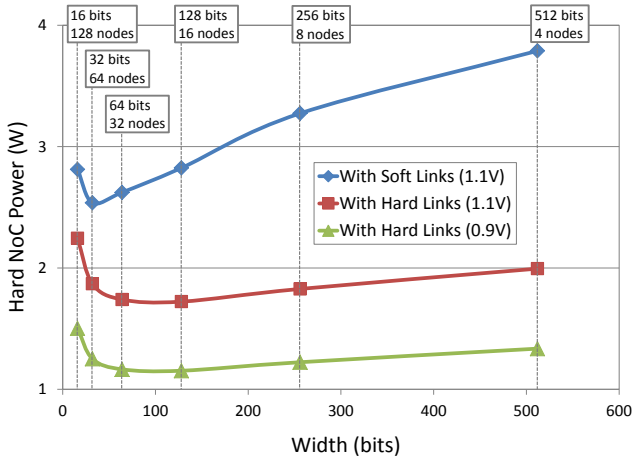


Fig. 10: Power of mixed and hard NoCs with varying width and number of routers at a constant aggregate bandwidth of 250 GB/s.

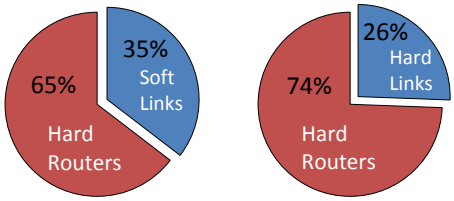


Fig. 11: Power percentage consumed routers and links in a 64-node mixed/hard mesh NoC.

V. Energy Efficiency of Complete NoCs

This section investigates the power consumed by complete NoCs, especially the mixed and hard NoCs presented in Section II. We investigate how the width of NoC links and spacing of NoC routers affect power consumption. Additionally, we report how much of the FPGA's power budget would be spent in these hard NoCs under worst-case traffic, if they are used for global communication.

We calculate the energy per unit of data moved by NoCs as an important figure of merit. This is used to compare the energy efficiency of different hard and soft NoCs. We also compare the energy per data of NoCs to conventional point-to-point links on the FPGA. Although point-to-point links merely connect two modules and are incapable of arbitration and switching between many nodes, this comparison shows how the presented NoCs compare to *best-case* conventional interconnect on the FPGA. We show that we can design a hard NoC that uses approximately the same energy as regular (soft) point-to-point links on the FPGA.

A. Power-Aware NoC Design

Fig. 10 shows the total dynamic power of mixed and hard NoCs as we vary the width. When we increase the width of our links we also reduce the number of routers in the NoCs to keep the aggregate bandwidth constant at 250 GB/s. For example, a 64-node NoC with 32-bit links has the same total bandwidth as a 32-node NoC with 64-bit links. However, with fewer routers the links become longer so that the whole FPGA area is still reachable through the NoC, albeit with coarser granularity. We assume that our NoCs are implemented on an FPGA chip whose core is 21 mm in each dimension as in the largest Stratix III device [22].

The power-optimal NoC link width varies by NoC type as Fig. 10 shows. The most power-efficient mixed NoC has 32-bit wide links and 64 nodes. However, for hard NoCs the optimum is at 128-bit width and 16 router nodes. The difference between the two NoC types is a result of the relative router:links power. With fewer but wider nodes, the total router power drops as the control logic power in each router is amortized over more width and hence more data. However, the link power increases since longer wires are used between the more sparsely distributed router nodes. Because soft links consume more power than hard links, they start to dominate total NoC power earlier than hard links as shown in Fig. 10.

Fig.11 shows the NoC power dissipated in routers compared to links for a 64-node NoC. On average, soft links consume 35% of total NoC power, while hard links consume 26%. For NoCs with fewer nodes (and hence longer links), the relative percentage of power in the links is higher.

B. FPGA Power Budget

We want to find the percentage of an FPGA's power budget that would be used for global data communication on a hard NoC. We model a typical, almost-full[4] FPGA using the Early Power Estimator [23]. The largest Stratix III FPGA core consumes 20.7 W of power in this case, divided into 17.4 W dynamic power and 3.3 W static power. Note that 57% of this power is in the interconnect, while 43% is consumed by logic,

TABLE III: System-level power, bandwidth and energy comparison of FPGA-based NoCs and regular point-to-point links.

FPGA-based NoCs

NoC Type	NoC Links	Description	Total Power	Aggregate Bandwidth	Energy per Data
Soft 64-NoC	Soft	1.1V, 167 MHz, 32 bits, 2 VCs	5.14 W	54.4 GB/s	94.5 mJ/GB
Mixed 64-NoC	Soft	1.1V, 730 MHz, 32 bits, 2 VCs	2.47 W	238 GB/s	10.4 mJ/GB
Hard 64-NoC	Hard	1.1V, 943 MHz[3], 32 bits, 2 VCs	2.67 W	307 GB/s	8.68 mJ/GB
Hard 64-NoC	Hard	0.9V, 943 MHz, 32 bits, 2 VCs	1.78 W	307 GB/s	5.78 mJ/GB
Hard 64-NoC	Hard	0.9V, 1035 MHz, 32 bits, 1 VC	1.21 W	236 GB/s	5.13 mJ/GB
Hard 64-NoC	Hard	0.9V, 957 MHz, 64 bits, 1 VC	1.95 W	437 GB/s	4.47 mJ/GB

[3]1.1 V routers can exceed 943 MHz as this freq. is achieved at 0.9 V.

Conventional Point-to-Point FPGA Interconnect

FPGA Interconnect Resource	Description	Total Power	Aggregate Bandwidth	Energy per Data
Equal use of C4,12 and R4,20	1.1V, 200 MHz, 10000 bits	1.18 mW	250 GB/s	4.73 mJ/GB

memory and DSP.

Aggregate (or total) bandwidth is the sum of available data bandwidth over all NoC links accounting for worst-case contention. A 64-node mixed NoC can move 250 GB/s around the FPGA chip using 2.6 W, or 15% of the typical large FPGA dynamic power budget of 17 W. A hard NoC is more efficient and consumes 1.9 W or 11% at 1.1 V and 1.3 W or 7% at 0.9 V. This implies that only 3-6% of the FPGA power budget is needed for each 100 GB/s of NoC communication bandwidth.

C. Comparing NoCs and FPGA Interconnect

We suggest the use of NoCs to implement global connections between compute modules on the FPGA; as such, we must compare to existing communication methods. There are two main types of interconnect that can be configured on the FPGA. The first uses only soft wires to implement a direct point-to-point connection between modules or to broadcast signals to multiple compute modules. The second type of interconnect is composed of wires, multiplexers and arbiters to construct buses. This is often used to connect multiple masters to a single slave, e.g. connecting multiple compute modules to external memory. In this section we compare our NoC power consumption with FPGA point-to-point links to get an indication of the "raw" efficiency of NoCs compared to this simple interconnect, and we compare NoCs to buses in Section VI.

The FPGA point-to-point links consist of a mixture of different FPGA wires that are equal in length to a single NoC link; 10,000 wires running at 200 MHz can provide a total bandwidth of 250 GB/s. We assume large packets on the NoC, so that the overhead of a packet header is negligible. Nevertheless, this comparison favors the FPGA links, because NoCs can move data anywhere on the chip as well as perform arbitration, while the direct links are limited in length to an NoC link and can perform no arbitration or switching.

Table III shows the result of this comparison. We start by looking at a completely soft NoC that can be configured on the FPGA without architectural changes. Under high traffic, this NoC consumes 5.1 W of power or approximately one third of the FPGA's power budget. However, because its clock frequency is only 167 MHz, it has a relatively low aggregate bandwidth of 54 GB/s. This means that moving 1 GB of data on this soft NoC costs 95 mJ of energy. Conventional point-to-point links only consume 4.7 mJ/GB; soft NoCs seem prohibitively more power-hungry in comparison.

Next, we look at mixed and hard NoCs. A mixed NoC is limited to 730 MHz because of the maximum speed of the FPGA interconnect; nevertheless, this is enough to push this NoC's aggregate bandwidth to 238 GB/s. Note that we calculate bandwidth from simulations and so we account for network contention in these bandwidth numbers. With hard routers and soft links, this NoC consumes 2.5 W or 10 mJ/GB, which is 2.2× that of point-to-point links.

A hard NoC can run as fast as the routers at 943 MHz raising the aggregate bandwidth to 307 GB/s. The energy per data for this NoC is 8.7 mJ/GB; 1.8× more than conventional FPGA links. In Section II we discussed that this completely hard NoC can run at a lower voltage than the FPGA. When looking at the same hard NoC running at 0.9 V instead of 1.1 V, the energy per data drops to 5.8 mJ/GB; 22% higher than conventional FPGA wires.

Next, we look at the overhead of VCs by investigating a one-VC version of our hard NoC running at 0.9 V. Some have suggested that VCs consume area and power excessively [6]. Table III confirms that supporting multiple VCs does reduce energy efficiency. Moving to one VC increases blocking at router ports, reducing aggregate bandwidth by 23% to 236 GB/s. However, power drops by 35% resulting in a reduced energy per data of only 5.1 mJ/GB, a mere 8% higher than the conventional FPGA wires.

Finally, by increasing the flit width of the NoC from 32 to 64 bits, we double its bandwidth while increasing power by only 61%. This increases energy efficiency to 4.5 mJ/GB, as the router control logic power is amortized over more data bits. This energy per data is *6% lower* than that of the conventional FPGA wires (4.7 mJ/GB).

These findings lead to two important conclusions. First, the most energy-efficient NoC avoids VCs, uses a wide flit width,

[4]Only core power is measured excluding any I/Os. We assume that our full FPGA runs at 200 MHz, has a 12.5% toggle rate, and is logic-limited. 90% of the logic is used, and 60% of the BRAMs and DSPs.

has hard links and a reduced operating voltage. Second, an embedded hard NoC with hard links on the FPGA can match or even exceed the energy efficiency of the simplest FPGA point-to-point links.

VI. Hard NoCs vs. Soft Buses

In this section we compare the efficiency of a hard NoC that is likely to be embedded on a high-performance FPGA to soft buses. Having compared the "raw" efficiency of NoCs against point-to-point links, we compare against buses of different parameters to understand exactly when a hard NoC is the better option.

We start by speculating on the hard NoC parameters that are likely to be used with FPGAs, motivated by I/O requirements and common FPGA micro-applications. Following that we investigate the efficiency parameters of soft buses that are currently used to interconnect systems on FPGAs. We use Qsys – a widely-used commercial system integration tool to generate these buses.

A. A Hard NoC for FPGAs

Even though the previous section looked at 32-bit 64-node NoCs from a "raw efficiency" perspective, we believe that the large number of nodes will be overkill for FPGA applications that typically have wide data-paths and few compute modules. Furthermore, a hard NoC must be able to interconnect important I/O and memory interfaces to the FPGA fabric; we look at three of these I/O interfaces on a 65-nm FPGA to motivate the parameters of a viable hard NoC.

1) DDRx Interfaces: Port width is typically 64 bits at double data rate (or 128 bits at single data rate), and it can run at 533 MHz or 800 MHz. The interface to the FPGA at full bandwidth is ~200 MHz and 512 bits wide.

2) PCIe Transceivers: A Gen-3 link can have 1, 2, 4 or 8 lanes each running at 8 Gb/s. An 8-lane interface to the FPGA would run at 250 MHz and be 256 bits wide.

3) Ethernet Ports: 10 Gb/s Ethernet is deserialized on FPGAs into a configurable-width datapath of up to 64 bits at ~150 MHz

TABLE IV: Hard NoC parameters suitable for an FPGA.

Size	Width	Area	Max. Frequency
16 nodes	128 bits	384 LBs	917 MHz

Of the three, the interface that requires the highest bandwidth is the DDR3 interface when running at full throughput. In this case a 32-bit wide NoC link is not enough to transport the bandwidth of DDR3; the DDR3 interface will have to be connected to more than one router port and the memory words will have to be segmented over the NoC then reassembled at their destination. Barring any such segmentation and re-coalescing of memory words, each NoC link must be able to transport the full bandwidth of DDR3 at 200 MHz $\times$ 512 bits $=$ 12.8 GB/s. Because hard NoCs can run at ~900MHz, we choose the NoC channel width of 128 bits such that the full data coming from DDR3 can be transported on a single NoC link. We therefore propose the parameters in Table IV; these parameters are the same as the power-optimal hard NoC parameters found in Fig. 10 as well. With these parameters, the area of this NoC is equivalent to 384 Stratix-III logic blocks, the energy per data is 7.65 mJ/GB and the frequency 917 MHz. We choose to run the NoC at the FPGA's nominal supply voltage (1.1 V).

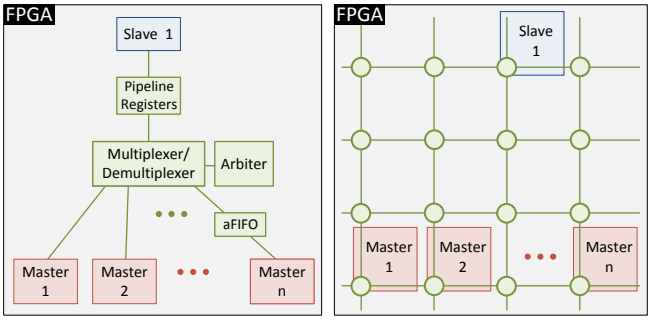


Fig. 12: Multiple masters accessing a single slave can be interconnected with a soft bus or hard NoC.

B. Multiple Masters Arbitrating for a Single Slave

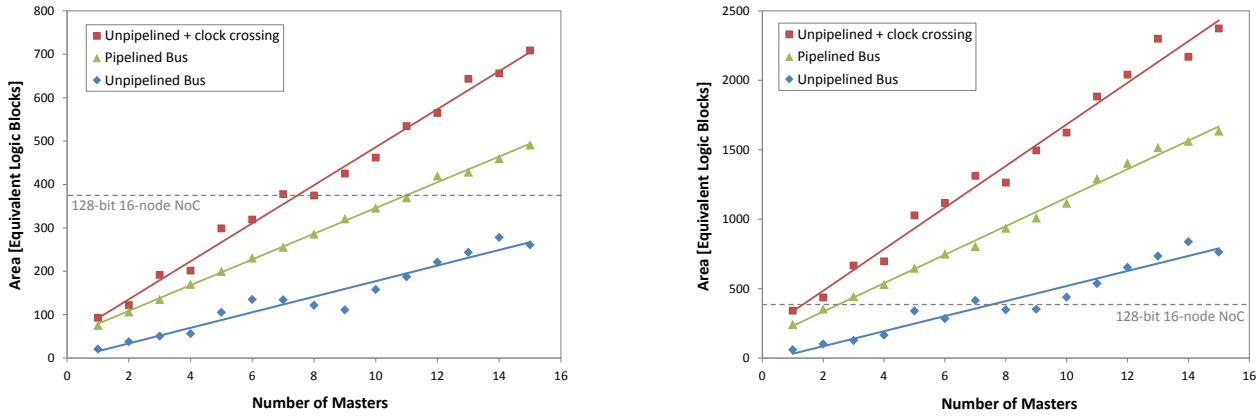
A common interconnection configuration is shown in Fig. 12 – it shows multiple masters connecting to a single slave through a multiplexer and arbiter, with optional pipeline registers and asynchronous FIFOs where clock domain crossing is necessary. An example of this bus configuration is when multiple modules are accessing memory; either an on-chip memory hierarchy or external memory such as DDRx memory.

Figures. 13–15 show the area, frequency and energy used by soft buses as compared to our hard NoC. We repeat the measurements for both 128-bit wide buses and 512 bits, and investigate both pipelined and unpipelined buses, and those with clock-domain crossing circuitry on half the masters. Note that hard NoCs already contain hardened clock-domain crossing circuitry at the input ports ("Fabric Interface" in Fig. 1) consisting of asynchronous FIFOs and multiplexers; this is to bridge between the FPGA clock domain and the NoC clock domain which will typically run at more than double the soft logic frequency [24].

1) Area:

We compute the physical chip area occupied by the NoC in equivalent logic blocks to be able to compare easily to the soft buses [24]. If a soft bus uses an FPGA block such as block RAM we compute the equivalent number of logic blocks that represent its area as well. Fig. 13 shows a comparison of hard NoC area with buses of width 128 bits (Fig. 13a) and 512 bits (Fig. 13b).

At 128 bits, the NoC area exceeds that of the unpipelined bus even for large systems with 15 masters and one slave. However, when the bus is pipelined, its area grows considerably, making the bus-based interconnect of an 11-master system almost as large as our high-bandwidth 128-bit hard NoC.



(a) 128-bit buses

(b) 512-bit buses

Fig. 13: Comparison of hard NoC area with (a) 128-bit buses and (b) 512-bit buses with different number of masters and a single slave.

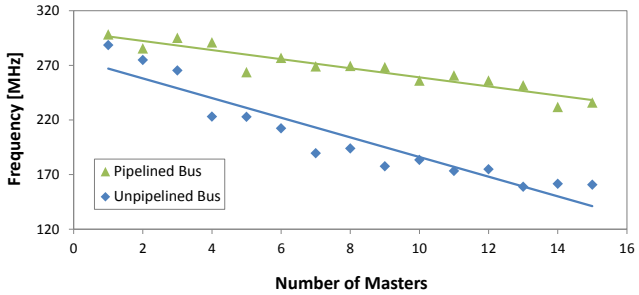


Fig. 14: Frequency of pipelined and unpipelined 128-bit buses with different number of masters and a single slave.

The third (red) curve shows the area of the bus-based interconnect when clock-crossing circuitry (mainly asynchronous FIFOs) are added to be able to connect modules of different clock domains together. That bloats area as these FIFOs are very area-expensive on FPGAs causing an 8-module system to be almost equally large as a hard NoC.

At 512-bits, buses become very large rather quickly with system size; with seven masters, even an unpipelined 512-bit bus is already as large as the NoC. However, the more relevant type of bus is the pipelined version; as Fig. 14 shows, pipelining can improve bus frequency by as much as 90 MHz. A pipelined 512-bit bus that connects 3 masters and one slave can run at ~240 MHz and is already larger than an full-fledged 128-bit hard NoC which runs faster than 900 MHz.

The area comparison indicates that a hard NoC is a viable replacement for buses in high-bandwidth applications; a narrow-but-fast hard NoC can replace traditional slow-but-wide soft buses on FPGAs for global communication.

2) Energy:

To be able to compare the power of hard NoCs and soft buses, we compute the energy required for a message to go from source to destination over both interconnect types. The energy-per-data metric introduced earlier is a quotient of power and aggregate bandwidth (sum of bandwidth of all links). Therefore, it pays no attention to where modules are located or how many hops a *message* must travel before reaching its destination. Effectively, it finds the energy for data to traverse one hop on the NoC – this was useful in comparing the raw energy of the NoC to wires of the same length as that one hop. However we now want to compare NoCs and buses, and must therefore find the energy of moving useful data between source and destination while taking into account the number of hops we traverse. For NoCs, we calculate this metric as follows:

$$\frac{Energy}{Message} = \frac{Energy}{Data} \times \#Hops \quad (1)$$

For unpipelined[5] buses, we find the energy-per-message by simulating the transfer of a number of messages from source to destination then dividing the measured power by the number of transfers-per-second. Fig. 15 shows the energy-per-message of NoCs compared to buses. The first observation of note is that the energy-per-message of wider buses is smaller than that of narrower buses; this is because the control logic for these soft buses is amortized over more data transferred per message in a wider bus. Secondly, the energy-per-message increases as we increase the number of masters connected through that bus. This is because the key bus components, such as multiplexers, become larger as we increase the number of modules connecting through the bus thus increasing capacitance and hence power. Additionally, as we connect more modules, they must be spaced out more, thereby increasing interconnect wire length and power.

In contrast, NoC energy-per-message decreases slightly in Fig. 15 since we compute the average energy of a message in the NoC when the modules are placed close together (and thus have fewer hops per message), and far apart with more hops between source and destination. This averaging causes the average number of hops per message in a smaller system

[5]We haven't reported the energy dissipation of pipelined buses because of a Quartus II software bug in the power breakdown by hierarchy, but it is significantly higher than that of unpipelined buses.

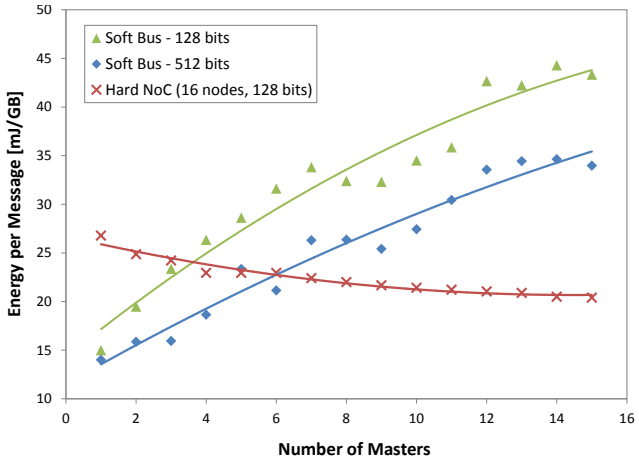


Fig. 15: Energy-per-data comparison of hard NoCs and unpipelined buses with different number of masters and a single slave.

to be larger than that of a larger system – this is made clearer in the following paragraph by looking at the data points.

On an NoC, if we have just one master and one slave, and they are placed one hop apart (best case), the energy required would be 7.65 mJ/GB, but if placed on opposite ends of the NoC with 6 hops in between them (worst case), 45.9 mJ/GB are required to transport that message – the average energy-per-message is therefore 26.8 mJ/GB. If we consider a system with 2 or more masters and repeat the computation of best-case and worst-case energy, we will find that it is slightly lower because the average number of hops from master to slave becomes less as well. For one master sending a 512-bit wide message to one slave over a soft bus, the energy dissipated is 14.1 mJ/GB, which is more efficient than our hard NoC.

As Fig. 15 shows however, the NoC becomes more energy-efficient for systems of 6 modules or more using a wide 512-bit bus, or systems larger than 3 modules using a 128-bit bus. Pipeline registers are often added to buses to improve their frequency as shown in Fig. 14, but this raises their energy consumption as well making NoCs an even more appealing energy-efficient alternative.

C. Example System Interconnect

We looked at how the efficiency of NoCs compare to a *single* soft bus as we vary its size. Furthermore, the hard NoC was very underutilized in these comparisons and can support much more communication bandwidth as we show in this section. However, FPGA systems typically have more than one bus to interconnect the modules in a system to each other, and to I/Os. We explore an FPGA system[6] that consists of two DDR3 interfaces running at 200 MHz and 512 bits connected in total to 7 on-chip modules, a link to an external device through PCIe and a control processor connected to all 8 modules. Table V lists the soft buses required to interconnect such a system, and the area and frequency of each bus.

We used Qsys to generate the buses and enabled pipelining only when necessary, for example, we needed pipeline registers to connect 5 modules to DDR3 memory and maintain a frequency higher than 200 MHz, while no pipeline stages were necessary in connecting a single module to the PCIe interface at 250 MHz. These frequencies (200 MHz for DDR3 and 250 MHz for PCIe) are the timing constraints for the respective interfaces and the bus must be designed to meet them. All the modules connected to that bus must either operate at that same frequency (200 MHz for DDR3 for example) unless we add clock crossing circuitry to the bus to operate a different (higher or lower) frequency. We assume that everything connected to the first DDR3 interface in Table V runs at 200 MHz, whereas the two modules connected to the second DDR3 interface do not run at exactly that frequency and hence require clock-crossing circuitry. Creating this clock-crossing circuitry out of soft logic consumes much area as demonstrated by Table V. Indeed, the soft bus that connects only 2 masters with clock-crossing is bigger than the one that connects 5 masters without clock-crossing. FPGA designs often use multiple clocks so we include clock crossing and width adaptation within our hard NoC in hard logic (hard logic is 30× smaller than soft logic for NoC components [2]).

To compute the aggregate bandwidth utilization of our system in Table V we assume that each interface is running at full bandwidth, so we compute the bandwidth as the product of bus width and frequency of each interface. However, the NoC aggregate bandwidth was pessimistically simulated under worst-case traffic as outlined in Section V. We also assume that modules will be placed on the NoC such that the distance traveled by data is average, or 4 hops. Each DDR3 interface supplies 12.8 GB/s, the PCIe link can transport 8 GB/s in each direction and the control processor requires 0.8 GB/s. When multiplying the total bandwidth by 4 hops, the aggregate amounts to 170 GB/s

As Table V shows, the summation of the bus areas for our sample system is 4× larger than our NoC even though the NoC is only 48% utilized when supporting the entirety of the system's communication – the area savings are significant. Such an embedded NoC can be used to interconnect the "infrastructure" of a system such as I/O interfaces with lower design effort as well. System designers currently struggle with these I/O interfaces to meet their stringent timing requirements and often need to repipeline their interconnect before arriving at a final design. However, by designing the NoC with these interfaces in mind, we can leverage the higher speed of the embedded NoCs in connecting these I/O interfaces with much lower effort.

It is of importance to consider latency – we largely leave this to future work. We expect the NoC to have a higher latency in "number of cycles"; however, since our NoC runs ~4.5× faster than soft buses (917 MHz vs. ~200 MHz), each cycle of latency on the NoC is much faster possibly leading to comparable latency in nanoseconds for both types

[6]The results in this section are generated by "Bus Designer": a fast prototyping tool for soft buses on FPGAs www.eecg.utoronto.ca/~mohamed/bus_designer.html

of interconnect.

VII. Conclusion

We studied how the power consumption of hard and soft NoC components varies with design parameters and data injection rates, and used that as the basis for designing energy-efficient NoCs. We presented mixed NoCs that use soft links to form an arbitrary topology and quantified their power consumption at ~6% of the FPGA's power budget for each 100 GB/s of data bandwidth. Hard NoCs consisting of hard routers and hard links are more power efficient, partially because they can be designed with a separate lower-voltage power grid. Our most power-efficient hard NoCs use only 4.5 mJ/GB to move data around an FPGA chip under high traffic, or ~3% of the FPGA power budget per 100 GB/s.

We then compared hard NoCs to the current form of interconnect on FPGAs; soft buses. Our high-throughput NoC is smaller and more energy efficient than a single 512-bit bus connecting 5 masters to 1 slave. When interconnecting a typical high-performance FPGA system, the NoC area is 4× smaller than soft buses with significant energy savings.

Acknowledgment

This work is funded by NSERC, Altera and Vanier CGS. Thanks to Daniel Becker for the open-source router, Natalie Enright Jerger, David Lewis, Dana How and Desh Singh for valuable discussions, and CMC for the ASIC CAD tools.

References

[1] Xilinx Inc., "Xilinx Ships First Virtex UltraScale FPGA and Expands Industry's Only 20nm High-End Family for 500G on a Single Chip," 2014. [Online]. Available: http://press.xilinx.com

[2] M. S. Abdelfattah and V. Betz, "Design Tradeoffs for Hard and Soft FPGA-based Networks-on-Chip," in *FPT*, 2012, pp. 95–103.

[3] E. S. Chung, J. C. Hoe, and K. Mai, "CoRAM: An In-Fabric Memory Architecture for FPGA-based Computing," in *FPGA*, 2011, pp. 97–106.

[4] B. Sethuraman *et al.*, "LiPaR: A Light-Weight Parallel Router for FPGA-based Networks-on-Chip," in *GLSVLSI*, 2005, pp. 452–457.

[5] M. K. Papamichael and J. C. Hoe, "CONNECT: Re-Examining Conventional Wisdom for Designing NoCs in the Context of FPGAs," in *FPGA*, 2012, pp. 37–46.

[6] Y. Huan and A. DeHon, "FPGA Optimized Packet-Switched NoC using Split and Merge Primitives," in *FPT*, 2012, pp. 47–52.

[7] R. Francis and S. Moore, "Exploring Hard and Soft Networks-on-Chip for FPGAs," in *FPT*, 2008, pp. 261–264.

[8] K. Goossens *et al.*, "Hardwired Networks on Chip in FPGAs to Unify Functional and Configuration Interconnects," in *NOCS*, 2008, pp. 45–54.

[9] G. Guindani *et al.*, "NoC Power Estimation at the RTL Abstraction Level," in *VLSI*, 2008, pp. 475 –478.

[10] R. Mullins, "Minimising Dynamic Power Consumption in On-Chip Networks," in *SoC*, 2006, pp. 1–4.

[11] H.-S. Wang, L.-S. Peh, and S. Malik, "A power model for routers: modeling Alpha 21364 and InfiniBand routers," *Micro*, vol. 23, no. 1, pp. 26–35, 2003.

[12] A. Sharifi *et al.*, "PEPON: Performance-Aware Hierarchical Power Budgeting for NoC Based Multicores," in *PACT*, 2012, pp. 65–74.

[13] A. Lambrechts *et al.*, "Power breakdown analysis for a heterogeneous NoC running a video application," in *ASAP*, 2005, pp. 179–184.

[14] F. Angiolini *et al.*, "Contrasting a NoC and a traditional interconnect fabric with layout awareness," in *DATE*, 2006, pp. 124–129.

[15] M. S. Abdelfattah and V. Betz, "The Power of Communication: Energy-Efficient NoCs for FPGAs," in *FPL*, 2013, pp. 1–8.

[16] Daniel U. Becker, "Efficient Microarchitecture for Network-on-Chip Router," Ph.D. dissertation, Stanford University, 2012.

[17] Altera Corp., "Stratix III FPGA: Lowest Power, Highest Performance 65-nm FPGA," Press Release, 2007.

[18] I. Kuon and J. Rose, "Measuring the Gap Between FPGAs and ASICs," *TCAD*, vol. 26, no. 2, pp. 203–215, 2007.

[19] J. Rabaey, A. Chandrakasan, and B. Nikolic, *Digital Integrated Circuits, A Design Perspective*, 2nd ed. Upper Saddle River, NJ: Pearson Education, Inc., 2003.

[20] W. Dally and B. Towles, "Route Packets, Not Wires: On-Chip Interconnection Networks," in *DAC*, 2001, pp. 684–689.

[21] W. J. Dally and B. Towles, *Principles and Practices of Interconnection Networks*. Boston, MA: Morgan Kaufmann Publishers, 2004.

[22] H. Wong, V. Betz, and J. Rose, "Comparing FPGA vs. Custom CMOS and the Impact on Processor Microarchitecture," in *FPGA*, 2011, pp. 5–14.

[23] Altera Corp., "Stratix PowerPlay Early Power Estimator." [Online]. Available:

[24] M. S. Abdelfattah and V. Betz, "Networks-on-Chip for FPGAs: Hard, Soft or Mixed?" in *TRETS*, 2014.

TABLE V: Interconnect efficiency comparison between a hard NoC and soft buses for a sample FPGA system.

Soft Buses for System Interconnect

Purpose	Bus Description	Area	Frequency
5 modules accessing full bandwidth of first DDR3 memory	5 masters, 1 slave, 512 bits, pipelined, no clock crossing	648 Logic Blocks	228 MHz
2 modules (different frequency) accessing full bandwidth of second DDR3 memory	2 masters, 1 slave, 512 bits, un-pipelined, clock crossing	654 Logic Blocks	219 MHz
PCIe link Gen3 8 lanes connected to one module	1 master, 1 slave, 256 bits, un-pipelined, no clock crossing	33 Logic Blocks	289 MHz
Soft processor connecting to the control port of the 8 modules	1 master, 8 slaves, 32 bits, un-pipelined, no clock crossing	97 Logic Blocks	234 MHz
Overall soft bus interconnect		1432 Logic Blocks	–

Hard NoC for System Interconnect

Description	Aggregate Bandwidth Utilization	Area Improvement over Soft Buses
16-node 128-bit NoC	$\frac{170\ GB/s}{352\ GB/s}$ = 48%	384 Logic Blocks = 3.7× (~4×) smaller