THE POWER OF COMMUNICATION: ENERGY-EFFICIENT NOCS FOR 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

Integrating networks-on-chip (NoCs) on FPGAs can improve device scalability and facilitate design by abstracting communication and simplifying timing closure, not only between modules in the FPGA fabric but also with large "hard" blocks such as high-speed I/O interfaces. We propose mixed and hard NoCs that add less than 1% area to large FPGAs and run 5-6× faster than the soft NoC equivalent. 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. In many designs, communication must include multiplexing, arbitration and/or pipelining. For all these cases, our results indicate that a hard NoC will be more energy efficient than the conventional FPGA fabric.

1. INTRODUCTION

FPGAs are becoming ever more capable devices, both by increasing in capacity and by integrating an ever more diverse set of hard blocks, such as high speed serial and memory interfaces and even complete processors. Though key to their success, these interfaces and embedded blocks are making it more difficult to design for FPGAs. It is challenging to meet the timing constraints and bandwidth needs of high-speed hard blocks using the FPGA's conventional interconnect, as buses that are both very wide and fast must be constructed. 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. Such buses can rapidly consume a large fraction of the FPGA resources, and they present a difficult CAD and timing closure challenge. 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, embedded blocks and the FPGA fabric (Fig. 1). The NoC abstraction can simplify design and speed up compilation [1, 2]. Our recent work showed
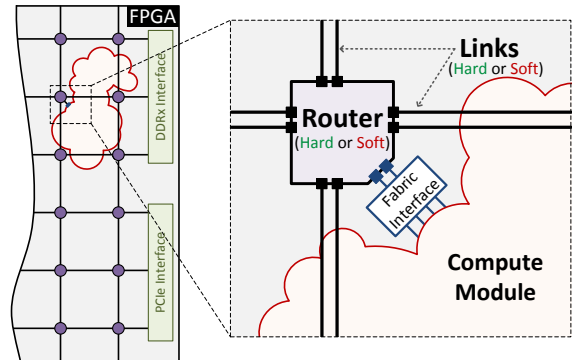
Fig. 1: A mesh NoC implemented on an FPGA. The example shows one router connected to a compute module and three links connected to each of the DDR and PCIe interfaces.

that hard NoCs have compelling area and delay advantages over soft NoCs [1]; however, power is a major concern: Does this higher level of interconnect abstraction come at an unacceptable power cost? In answering this question, 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.

Both soft NoCs [3–5] and hard NoCs [6, 7] 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 [8–10]. Other work focuses on complete systems and reports the power budgeted for communication using an NoC [11, 12]. 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 [13]. 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 NoCs. We start by looking at the power consumption of each NoC component, both when implemented hard and soft, and how each component's power consumption varies with different design parameters. We then 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. Finally, we show that a hard
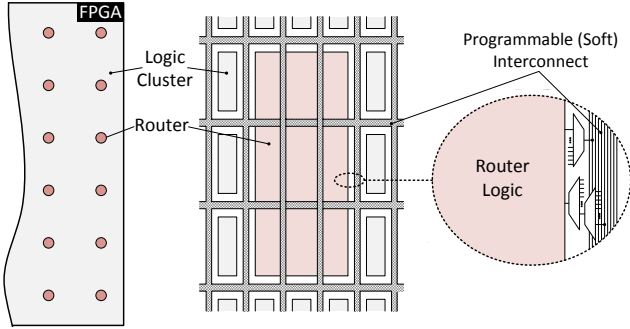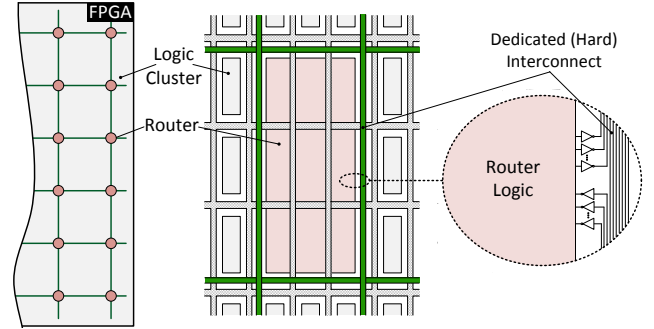
Fig. 2: Floor plan of a hard router with soft links embedded in the FPGA fabric. Drawn to a realistic scale.
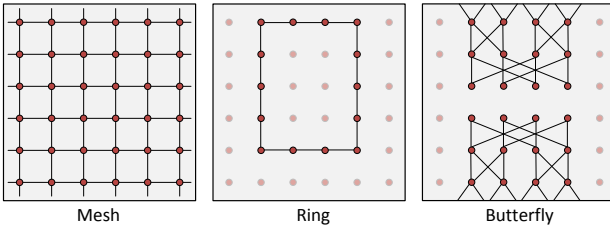

Fig. 3: Examples of different topologies that can be implemented using the soft links in a mixed NoC.

NoC can be as energy-efficient as point-to-point soft links on an FPGA. Point-to-point soft links cannot perform arbitration and switching; nevertheless, hard NoCs can be as power efficient as this simplest form of FPGA interconnect, proving that hard NoCs are not only area efficient and fast [1], but power efficient as well. Our contributions include:

- 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.

2. 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.

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 [14]. 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 [1, 14].

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

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

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 [1].

2.1. 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 $20\times$ smaller and $5\times$ faster than a soft NoC [1]. 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.

2.2. 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 $23\times$ smaller and $6\times$ 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.

3. 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 [15].

We start with an NoC with the baseline router parameters listed in Table 1. 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 [16].

Table 1: Baseline router parameters.

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

3.1. 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 [1]. 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.

3.2. Links Power

3.2.1. 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.

3.2.2. 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 [17]. Propagation delay is measured for both rising and falling edges of a square pulse signal, and the worst case

Table 2: 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

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 μm, 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 [18].

4. POWER ANALYSIS

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.[1] 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.

4.1. Router Power Analysis

4.1.1. Router Dynamic Power Ratios

As Table 2 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 [1]. 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 gap triples from 50$\times$ at six or less ports, to 150$\times$ 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 [19]. 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.

4.1.2. 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 [8, 13], 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 [1]. 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

---

[1]To access and visualize our complete area/delay/power results, please visit: `www.eecg.utoronto.ca/~mohamed/noc_designer.html`
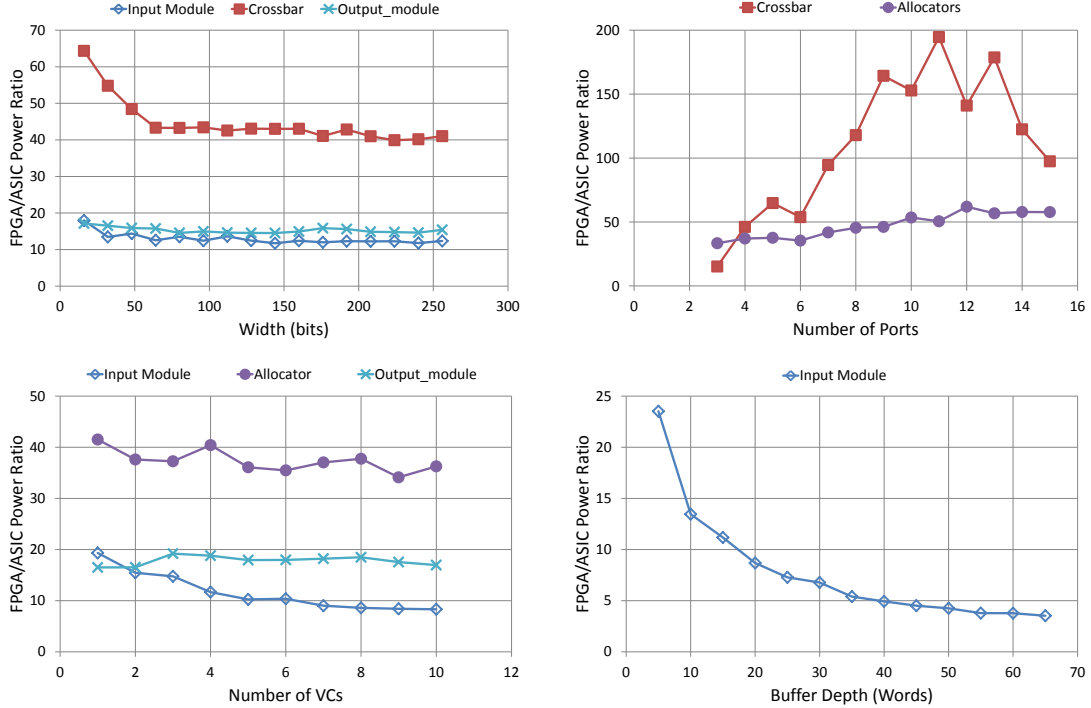
Fig. 5: FPGA/ASIC (soft/hard) power ratios as a function of key router parameters.

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.

4.1.3. 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 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 [9]. 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 [19].
- **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.

4.2. 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
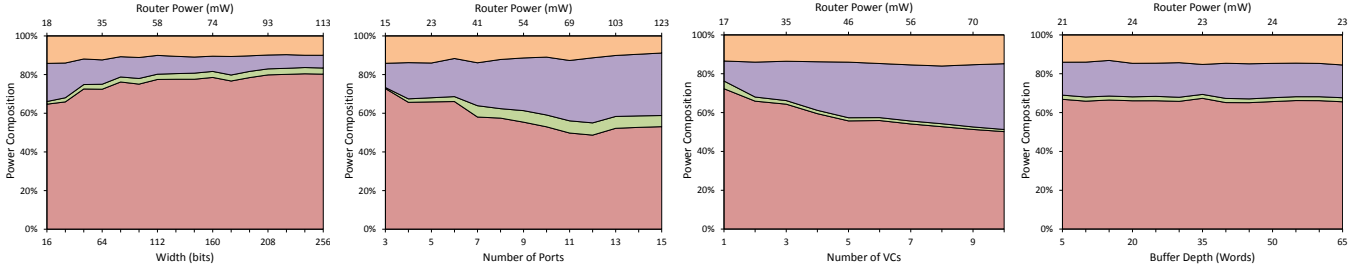
Fig. 6: FPGA (soft) router power composition by component and total router power at 50 MHz. Starting from the bottom (red): Input modules, crossbar, allocators and output modules.
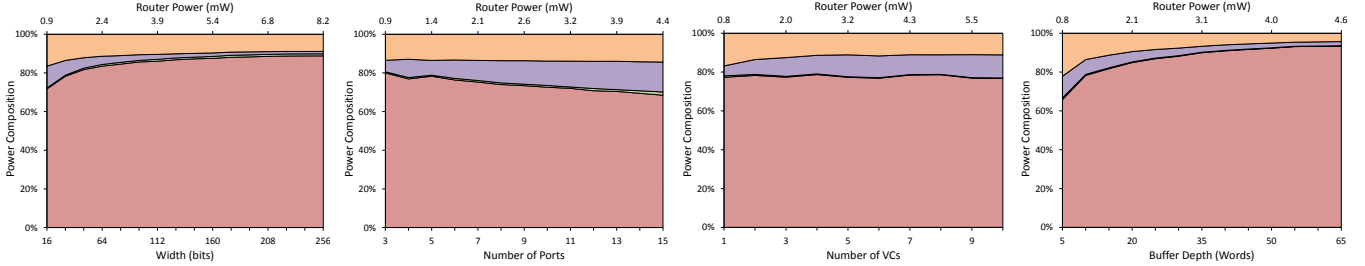

Fig. 7: ASIC (hard) router power composition by component and total router power at 50 MHz. Starting from the bottom (red): Input modules, crossbar (very small), allocators and output modules.
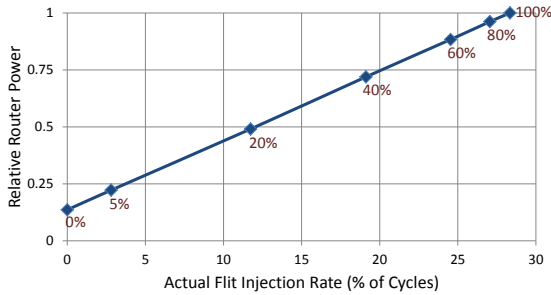

Fig. 8: Baseline router power at actual data injection rates relative to the its power at maximum data injection. Attempted data injection is annotated on the plot.

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.

5. SYSTEM-LEVEL COMPARISON

This section investigates the power consumed by complete NoCs, especially the mixed and hard NoCs presented in Section 2. 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.

5.1. 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 [20].

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.

5.2. 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[2] FPGA using the Early
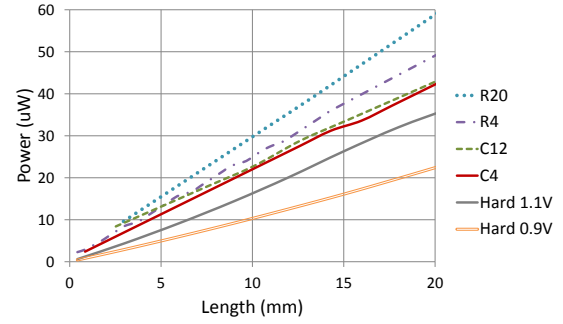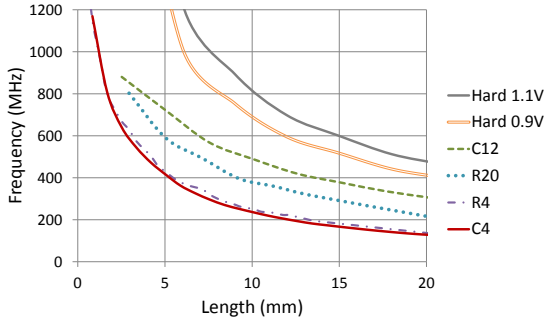
Fig. 9: Hard and soft interconnect wires frequency, and power at 50 MHz and 15% toggle rate.
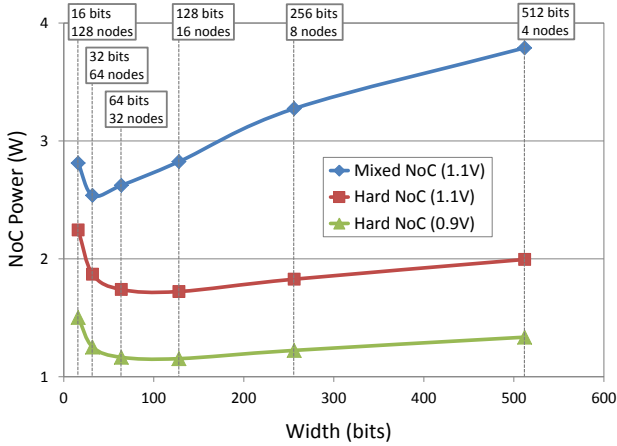

Fig. 10: Power of mixed and hard NoCs with varying width and number of routers at a constant aggregate bandwidth of 250 GB/s.
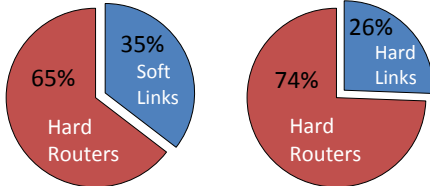

Fig. 11: Power percentage consumed routers and links in a 64-node mixed/hard mesh NoC.

Power Estimator [21]. 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, 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.

To put this in context, 250 GB/s is a large aggregate bandwidth. A single 64-bit DDR3 interface running at the current maximum frequency supported by any FPGA of 933 MHz,

produces a maximum data rate of 14.6 GB/s. A PCIe Gen3 x8 interface produces 8.5 GB/s of data in each direction. If this data is transferred to various masters and slaves located throughout the entire FPGA, the average distance traveled is half the width or height of the chip, or 4 routers. Hence an aggregate NoC bandwidth of $(14.6\times4)+(8.5\times2\times4)=126$ GB/s can distribute the maximum data from these high-speed interfaces throughout the entire FPGA chip.

5.3. 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. Although the proposed NoCs can implement both of these communication requirements (point-to-point and arbitration), we compare our NoC power consumption with the simplest FPGA point-to-point links. 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 3 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

---

[2]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.

Table 3: System-level power, bandwidth and energy comparison of different 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

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

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 2 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 [5]. Table 3 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, 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. This means that a hard NoC, integrated within the FPGA fabric, can implement global communication more efficiently than any soft interconnect that includes arbitration and switching. Hard NoCs are not only area-efficient and fast [1], but energy efficient as well.

6. 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. This is less than the energy required to move data on point-to-point soft links that are incapable of arbitration or switching, indicating that hard NoCs can result in overall power savings for FPGAs.

REFERENCES

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

[2] E. S. Chung, *et al.*, "CoRAM: An In-Fabric Memory Architecture for FPGA-based Computing," *FPGA*, 2011, pp. 97–106.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

[17] J. Rabaey, *et al.*, *Digital Integrated Circuits, A Design Perspective*, 2nd ed. Upper Saddle River, NJ: Pearson Education, Inc., 2003.

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

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

[20] H. Wong, *et al.*, "Comparing FPGA vs. Custom CMOS and the Impact on Processor Microarchitecture," *FPGA*, 2011, pp. 5–14.

[21] Altera Corp., "Stratix PowerPlay Early Power Estimator." [Online].

---

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