Power Reduction Through Physical Placement of Asynchronous Routers

Daniel Gebhardt Kenneth Stevens University of Utah

Contribution

This work reduces power consumption of our unique asynchronous NoC by minimizing path lengths and hop-count using force-directed and simulated annealing algorithms. We describe techniques for generating an optimized tree topology and router placement for power-constrained fixed-function SoCs composed of soft-IP blocks.

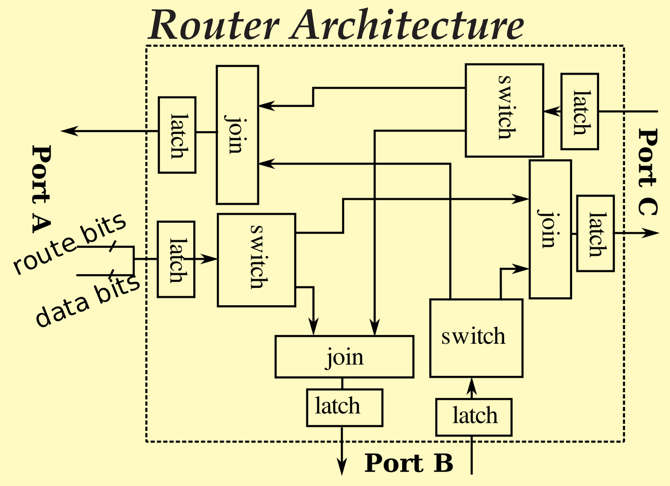
Motivation

Little existing research provides design automation algorithms for asynchronous NoCs, which show promise for power-constrained designs. Async NoCs have a number of advantages over clocked networks including a lack of a power-hungry clock tree and "automatic" clock gating to inactive routers. **Long wires** composing NoC links are already a large power consumer, and will increase relative to transistors in smaller process technologies. **Buffer power** will also get worse as links require additional pipelining. These two issues drive our network design and optimization methods.

Network

We consider an async network composed of simple, **three bidirectional port** (binary choice) routers. The simplicity of the design produces a **small router** that will give more flexibility to physical placement **within soft-IP** blocks. This allows for **reduced link length** in a **tree topology**, which exploits communication locality and simplifies routing. **Source-routing** bits are attached to each **1-flit packet**. This arrangement reduces blocking from long flit-trains. **I/O buffers** are single **latches** on input and output. The switching logic is **2-to-1 MUXs**, shown as the *join* module below. We use a **bundled-data** link protocol which is significantly more efficient on long links than latency-insensitive protocols.

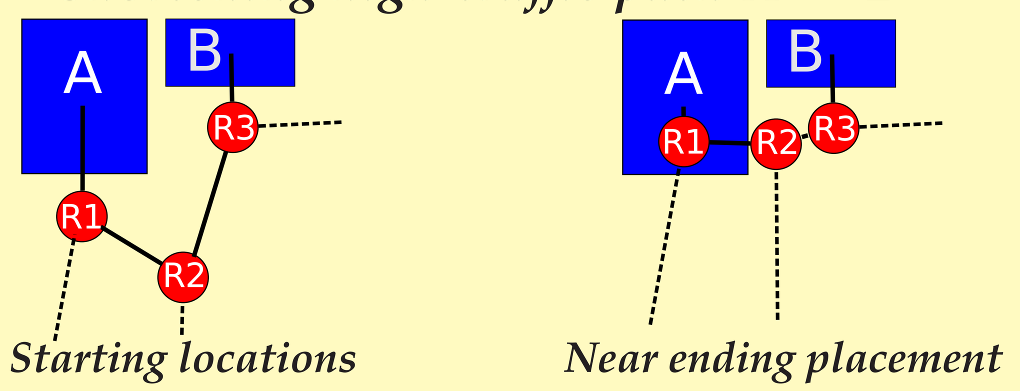
Router Architecture



Force-directed Router Placement

This method **minimizes the physical distance** that packets travel on **highly trafficked paths**, thus reducing the energy required to send data across them. Our method applies "force" to a router that **moves the router on the floorplan** to shorten a particular source-to-destination path. The magnitude of this force is proportional to the **amount of traffic** a path carries and its **wirelength**.

Shortening high-traffic path $A \rightarrow B$

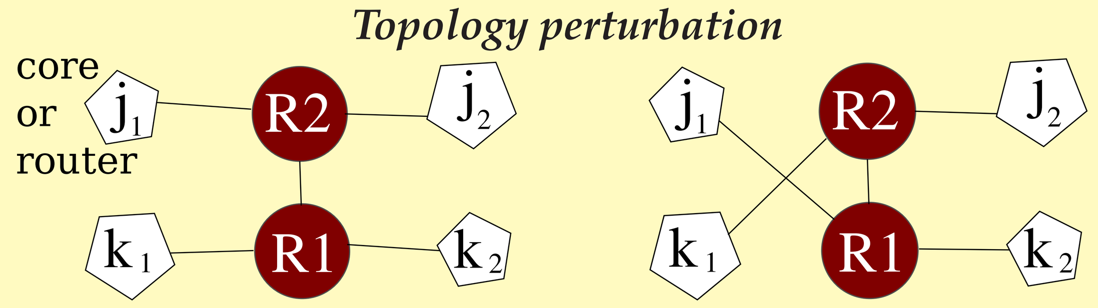


The **inputs** to this algorithm are the IP core **floorplan**, the network **topology**, and a **Core Communication Graph** expressing the average bandwidth between any two cores, as in other similar work [2]. We use the core center for distance calculations in this work, but could use the location of the network adapter.

Topology Generation

We use **simulated annealing** (SA) to search for a tree-based topology and router placement with minimal *fitness*. A solution's **fitness** is based on the summation of each path's **wirelength** and **router hop count**, each multiplied by a weighting factor representing **traffic volume** on that path. The initial topology is a balanced binary tree where frequently communicating cores are placed near each other. A neighbor-state is selected by perturbing the topology, but maintaining the tree requirement. Only when weighted hop-count improves is force-directed router placement performed, the candidate's fitness calculated, and the new solution accepted as per SA. This process repeats until "cooled."

Topology perturbation

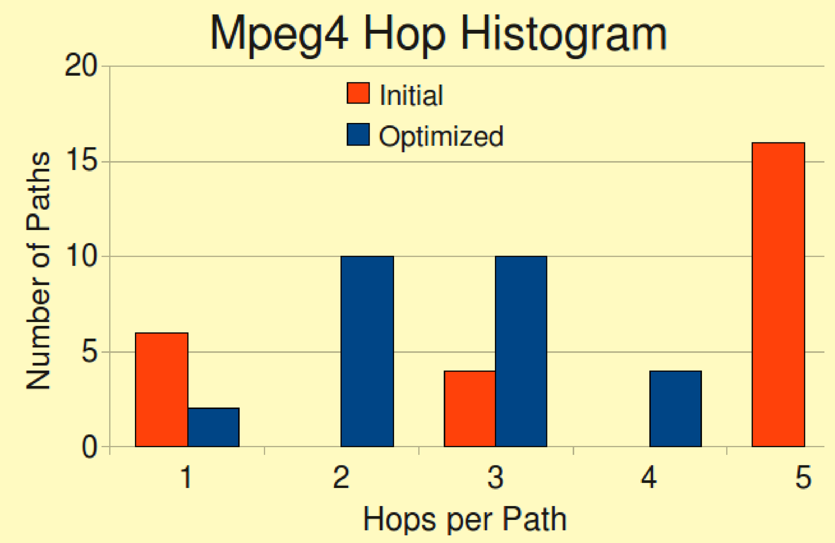


Future work will incorporate heuristics that make a better guess for neighbor-state selection. This will reduce runtime and improve quality.

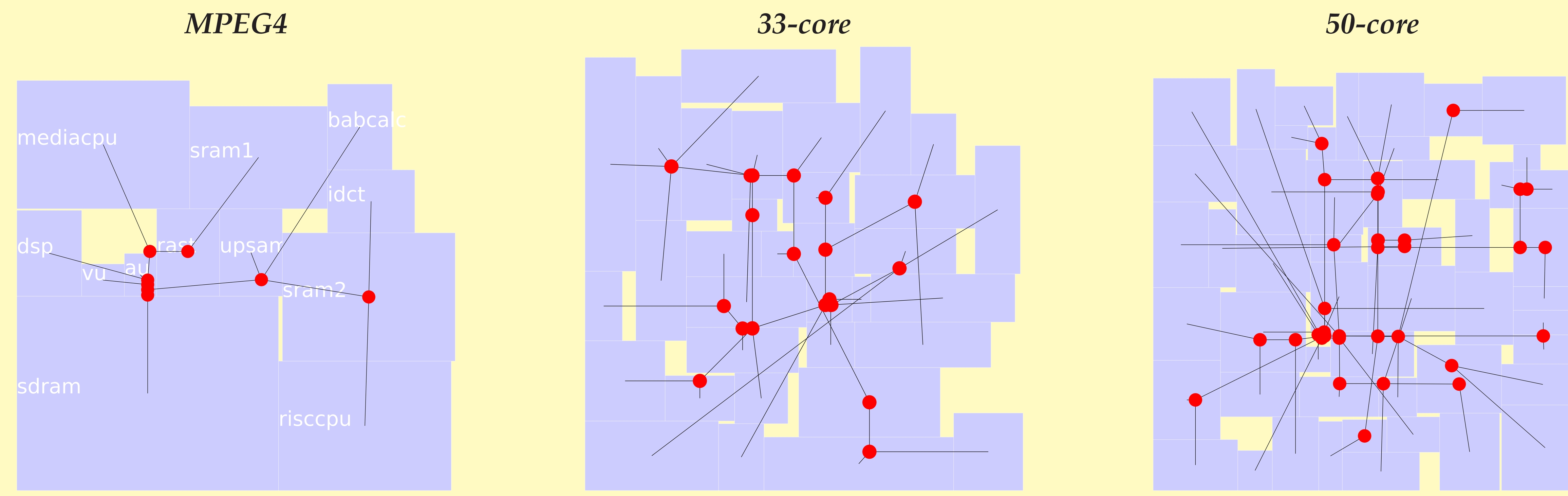
Evaluation & Results

We evaluated this work by measuring the **fitness improvement from the initial solution to the final post-SA solution**. The initial topology is a balanced tree, where a heuristic maps cores to leaves [1]. Force-directed placement is done to complete the initial solution. Three sample SoCs were used: a 12-core MPEG4 decoder, a 33-core, and a 50 core SoC. The latter two use synthetically generated traffic patterns to estimate a design. The floorplans were generated with Parquet and forced to a square aspect ratio. We vary a parameter that specifies the influence on solution fitness that wirelength has compared to router hops (WL:Hops). It represents the relative energy cost of a router vs. link and shows the sensitivity of this methodology to a range of implementation details, such as wire repeater sizing and spacing. The results show a gently **increasing benefit when wire energy costs increase**.

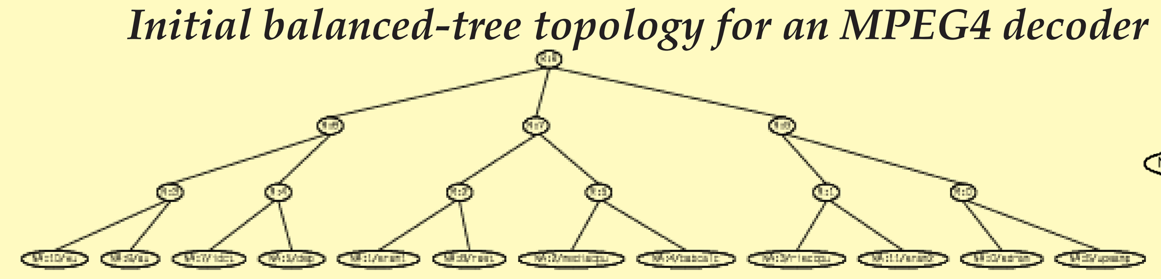
	WL:Hop Fit. Influence			
	1:10	1:1	10:1	
Design	% Fit. Imprv.			Time(min)
MPEG4	32	40	47	1
33-core	27	26	30	100
50-core	17	19	24	250



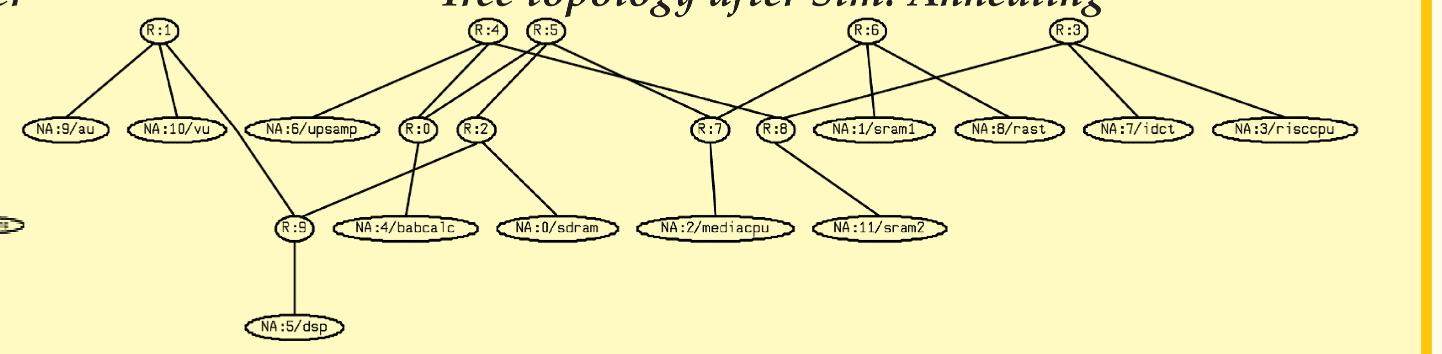
Router locations on the floorplans are shown below. Routers that overlap cores may be moved slightly to legalize violations (even in soft-IP) during final cell placement. **Clusters** of routers form when their connected cores have high cross-traffic. This is desired, as the **latency through the cluster** would be less than an entire clock cycle for each switch traversal in a synchronous NoC. These clusters are abstractly similar to a **finely clock-gated switch** with more ports, but suffer from congestion unlike a true crossbar switch. We will further investigate this clustering effect, and perhaps use it to determine **where to place higher-radix routers**.



Initial balanced-tree topology for an MPEG4 decoder



Tree topology after Sim. Annealing



References

[1] D. Gebhardt, K. Stevens. Elastic Flow in an Application Specific Network-on-Chip in *Wkshp on Formal Methods in GALS 2007*

[2] S. Murali, et al. Designing Application-specific networks-on-chips with Floorplan Information in *ICCAD 2006*

Funding

This work was supported by a grant from the United States National Science Foundation.