C-SURF
Computer Science Undergraduate Research Fellows Program

Mapping Circuits to Self-Assembled Networks

Cheney Tsai (cct8@cs.duke.edu)

Introduction

Recent research in novel nanotechnologies has led to the generation of resource-constrained self-organizing nano-scale networks. Using the context of the Self-Organizing SIMD Architecture(SOSA), we aim to explore the possibility of mapping circuit netlists onto these irregular, highly defective structures[1].

We intend to determine if current toolchains such as Quartus, the RASP package developed by the University of Berkeley, the VPR tool developed by the University of Toronto can be modified to adapt to this process, and identify the variables involved including: netlist size, architecture topology, node fanout, and others.

Background

SOSA is a defect-tolerant self-organizing nano-scale SIMD architecture built from a randomized network of simple nodes with capabilities comparable to that of a 1-bit ALU. Self-assembly can scale to 10^{12} node arbitrary network and provides optional control for node placement, node orientation and link growth.

We assume an assembly process to replicate a simple circuit cell consisting of a transistor in the cavity of a DNA-lattice[2].The primary netlists used are from the ISCAS-85/89 and 74X series. We'll also assume a 1:1 gate:node mapping.

Related Work

Existing systems that may be modified to place and rout onto irregular nano-scale networks include:

Teramac Project (HP 1998): Experimental massively parallel computer composed of defective Field-programmable Gate Arrays(FPGA).

RASP (UCLA VLSI CAD LAB 2004): An FPGA/CPLD technology mapping and synthesis package

Quartus University Interface Program Toolkit(Altera 2011): Open Quartus project flow allowing modifications from HDL synthesis, gate decomposition to LUT mapping, and routing.

VPR and T-Vpack (UofToronto, 2000): Versatile packing, placement and routing for FPGAs

Figures

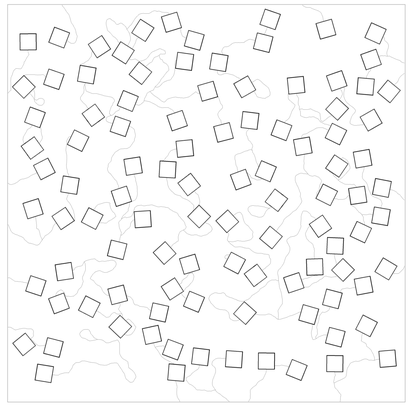

Figure 1: 100 Node SOSA Architecture with fanout of max. 3

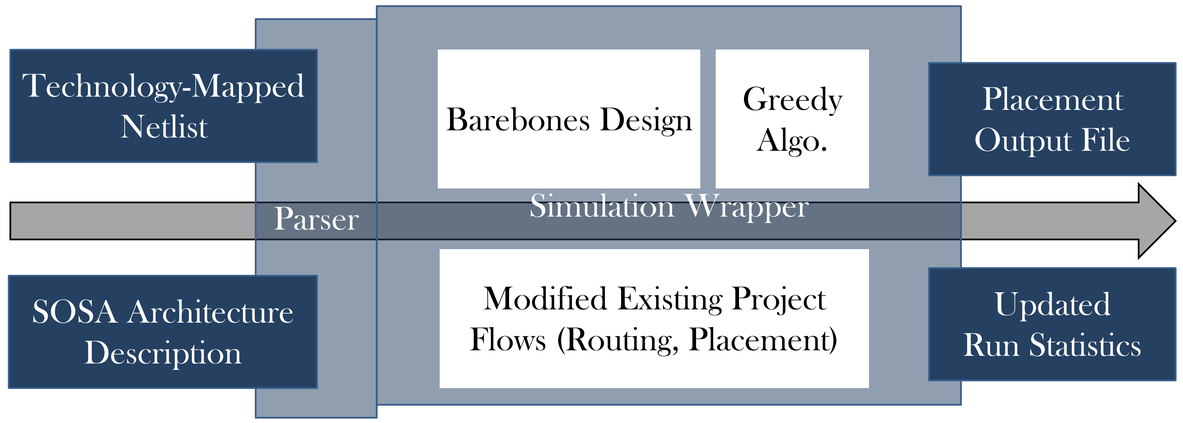

Figure 2: Block diagram of proposed simulation system flowing from left to right

Simulation Design & Analysis

Barebones Design

We define both the circuit netlist and the architecture topology as hypergraphs of vertices X, and edges E, where an edge can connect any number of vertices:

$$X = \{ x_m | m \in \mathrm{M} \} \qquad E = \{ e_i | i \in I, e_i \subseteq X \}$$

Minimizing critical path in a successful mapping is thus NP-complete in such a scenario. We begin with a greedy implementation using a Monte Carlo method to determine the initial placement. Traversing the netlist with a greedy breadth-first search yielded less backtracking for higher-fan out topologies. Also, pass-through nodes, high-fan out, logic decomposition, and more were necessary to handle edge cases.

Simulation failed to complete without allowing the cutting of interconnects instead of disabling collateral nodes of a successful node mapping. There was not enough time to construct and customize the implementations of the algorithms developed for FPGA placement and routing. Overall, mappings by hand and initial greedy simulations return few successful mappings.

Repurposing of Existing Tools

We hypothesize the possibility of modifying existing systems to simulate the mapping onto irregular networks which is similar to mapping onto highly defective FPGAs. In 2 cases(QUIP & RASP), the exact architecture could not be modified to use the outputs from SOSA. UofToronto's VPR and T-Vpack should be able to provide for that modification, which remains to be done.

Results

Netlist	# of Nodes In Topology: 10	30	120	200	250	300
Full Adder (5 gate elements)	>20%	>40%	>50%	>50%	>50%	>50%
S208.1isc (66 gate elements)	NA	NA	<10%	<10%	<20%	<20%
74L85.isc (188 gate elements)	NA	NA	NA	<10%	<10%	<10%

Table 1: Approximate successful placements from hand tracing and computer-aided simulation. (Out of 10 trials)

Conclusions and Extensions

We managed to explore the difficulties involved in mapping circuits onto irregular networks. While altering parameters like maximum fanout, node function, and link disabling may improve the success ratio of mappings, it is difficult to fully control these parameters at the nano-scale. We have also shown that a greedy approach has a very low success ratio.

Current routing and placement technologies employ many sophisticated algorithms that, while applicable, are optimized to regular networks, and not completely suitable for irregular, highly defective graphs. The next steps should involve more computationally rigorous simulations of circuit mappings to find better correlations between the success ratio and the network generation parameters. A project flow to compare this statistic among varying decomposition, routing, and placement algorithms would be essential to determine their viability.

DUKE COMPUTER SCIENCE

Special Thanks To:
Dr. Alvin Lebeck, Yang Liu
Duke CS Department

References:
[1] Y. Liu, C. Dwyer, A. Lebeck. *Routing in Self-Organizing Nano-Scale Irregular Networks.* ACM JETC, vol. 6, no. 1, pp. 1-21, 2010
[2] J.P., Patwardhan, C. Dwyer, A, Lebeck, *A Defect Tolerant Self-Organizing nanoscale SIMD architecture,* Proceedings of the 12th International Conference on Architectural Support for Programming Languages and Operating Systems. 2006c.