Liquid: Fast Placement Prototyping Through Steepest Gradient Descent Movement

Elias Vansteenkiste
Seppe Lenders
Dirk Stroobandt
Ghent University
Problem: Slow FPGA design cycle

(Re-)designing accelerator

Compilation

Check constraints

Design cycle

Runtime breakdown

Vivado - VTR benchmark designs

Example: Bitcoin miner design - 1M blocks:
Quartus II placement tool requires 20 minutes.
Current solution: Multi-threaded versions of existing approaches

Two common placement techniques:

- Simulated annealing
- Analytical placement

Not designed to exploit the high number of cores in GPU accelerators

Design new placement technique based on analytical placement
Analytical Placement: largest runtime consumer?

**Analytical placement cycle**

- Build linear system
- Solve
- Legalize
- Solve linear system
- Optimize system

Stop condition: solved solution cost reaches fraction of legal solution cost

Is it necessary to exactly solve the linear system? Legalization partly destroys the solution anyway.

**Runtime breakdown analytical placement**

- Initialize data: 16.4%
- Build linear system: 9.5%
- Solve linear system: 68.5%
- Calculate cost: 2.3%
- Legalization: 1.7%

Bitcoin miner design
Liquid: Iterative SGD optimization

- Optimize system by moving each block several times in the direction which reduces the cost the most.
- Momentum simulation to smooth out and make less prone to local minima.

\[ \mathbf{v}_i = \mu \cdot \mathbf{v}_{i-1} - \gamma \cdot \nabla C(x') \]
\[ x' = x_{i-1} + \mu \cdot \mathbf{v}_{i-1} \]
\[ x_i = x' + \mathbf{v}_i \]
Liquid: implementation details

Slight change in how the system is built:

● Each net: spring between extreme clusters
● Long springs pull harder
● Extra spring for highly critical source-sink connections

Combining spring forces:
Effort level

Number of gradient descent iterations before legalization

Decreases as placement progresses
Results
Conclusion and Future work