SNS’s Not a Synthesizer: A Deep-Learning-Based Synthesis Predictor

Ceyu Xu  
ceyu.xu@duke.edu  
Duke University

Chris Kjellqvist  
christopher.kjellqvist@duke.edu  
Duke University

Lisa Wu Wills  
lisa@cs.duke.edu  
Duke University
Motivation: Synthesis is Slow!

• Synthesis is an important source of feedback during the hardware design process
  • Estimates of a design’s physical characteristics
  • High-dimensional **Design Space Exploration** (DSE)

• Synthesis for a small 32x32 float32 systolic array can take up to one day (Die area around 2.5mm^2 with a 15nm library)
  • DSEs are computationally infeasible

• Neural Networks can estimate synthesis result much faster!
module MAC(
  input [7:0] a,
  input [7:0] b,
  output [15:0] y);

reg [15:0] acc;
assign y = acc;

always@(posedge clk)
begin
  acc <= acc + a * b;
end

...
Each path starts and ends with either an io-port or a D-flip-flop.

- [io8, mul16, add16, dff16]
- [io8, mul16, add16, dff16]
- [dff16, add16, dff16]
Paths
- [io8, mul16, add16, dff16]
- [io8, mul16, add16, dff16]
- [dff16, add16, dff16]

Power, Timing, Area
- 0.4ns, ...
- 0.4ns, ...
- 0.3ns, ...

Linear Regression?

- add16, mul16
- mul16, add16

Synthesis
Separate modules
is larger than
Fused MAC Unit
## Circuitformer

- “Attention” on...
  - Adjacency
  - Ordering
  - Frequency

<table>
<thead>
<tr>
<th></th>
<th>Circuitformer</th>
<th>Transformer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vocab Size</td>
<td>79</td>
<td>30522</td>
</tr>
<tr>
<td>Encoder Layers</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>Attention Heads</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>Embedding Dim</td>
<td>128</td>
<td>768</td>
</tr>
<tr>
<td>Input Length</td>
<td>512</td>
<td>512</td>
</tr>
<tr>
<td>#Parameters</td>
<td>1.4M</td>
<td>109M</td>
</tr>
<tr>
<td><strong>Token/Input</strong></td>
<td>Circuit</td>
<td>Vocabularies in English</td>
</tr>
<tr>
<td></td>
<td>functional</td>
<td></td>
</tr>
<tr>
<td></td>
<td>units</td>
<td></td>
</tr>
<tr>
<td><strong>Prediction</strong></td>
<td>Circuit path</td>
<td>Meaning of a sentence</td>
</tr>
<tr>
<td></td>
<td>PPA</td>
<td></td>
</tr>
</tbody>
</table>
A small multi-level perceptron is enough!
Circuit details have been captured by the circuitformer model.
Circuit Level Dataset: Collection

1. Collect Hardware Designs
2. Synthesis these designs using Synopsys Design Compiler and FreePDK15(OCL) library
3. Construct the Dataset.
Path Level Dataset: Augmentation

1. Obtain as many hardware designs as possible under **data scarcity**
2. Paths are sampled from hardware designs.
3. SeqGAN is used to generate mutations.
4. All the paths are synthesized using the same process.

- Generate Mutation of Path-set.
- Robustness++
- Accuracy++
SNS and Prior Works

- GNNs are accurate but scale poorly with memory
  - D-SAGE is a GraphSage model for predicting design timing
    - Lacks scalability
  - GraphSage consumes intractable amounts of memory and is unreasonably slow for large circuits (>1M vertices)

- SNS uses **Path Based Method (PBA)**
  - DeepWalk uses path-based methods to perform graph analytics
  - Maps well to circuit inference because it easily captures the one-cycle behavior of circuits
  - **Scales very well!** SNS’s path-based model scales to quickly produces inferences for hardware designs with up to 67.8 millions of transistors

*Speed is the Key!*
Accuracy

- Train/test using 50% of the dataset each
- Measure Root Relative Square Error (RRSE)

\[ RRSE = \sqrt{\frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2}} \]

<table>
<thead>
<tr>
<th></th>
<th>SNS</th>
<th>D-SAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timing RRSE</td>
<td>0.67</td>
<td>0.83</td>
</tr>
<tr>
<td>Power RRSE</td>
<td>0.60</td>
<td>-</td>
</tr>
<tr>
<td>Area RRSE</td>
<td>0.22</td>
<td>-</td>
</tr>
</tbody>
</table>
Performance

- Scale to Large circuits
- Larger circuit shows greater speedup
- 760x speedup on average
General-Purpose Core: BOOM

1. L1 Instruction Cache
   - 32*-KIB
   - 8*-way

2. Fetch-Target-Queue
   - (32*-entry)

3. BTB*
   - (1-cycle redirect)

4. Gshare* BPU
   - (3-cycle redirect)

5. Return Address Stack (RAS)

6. Fetch Buffer
   - (32* entries)

7. Instruction Fetch & PreDecode
   - (4 cycles)
   - (16* Byte window)

8. FrontEnd
   - ICache TLB*
   - ICache Tags*

9. 16 Bytes/cycle

10. 4*-Wide Decode
    - Decoder

2592 design points in total
Takes 45 days to synthesis
SNS takes 2.1 hours
Pareto Frontier of BOOM DSE shows "diminishing single core return"
Conclusion

- SNS is a tool for estimating the physical properties of hardware designs
  - High accuracy than prior work
  - On average 760 X faster than traditional synthesis
  - Enables high-dimensional design space explorations for large-scale designs