LAMDA: Learning-Assisted Multi-Stage Autotuning for FPGA Design Closure

Ecenur Ustun, Shaojie Xiang, Jinny Gui, Cunxi Yu, Zhiru Zhang

Electrical and Computer Engineering, Cornell University
Auto-Configuring FPGA CAD Tools

Timing distribution for various tool configurations and timing constraints

- InTime [Kapre et al., FPGA’15] [Yanghua et al., FPL’16]
- DATuner [Xu et al., FPGA’17]
- Nautilus [Papamichael et al., DAC’15]
Leveraging Multi-Stage ML Inference

C/C++/OpenCL → High-Level Synthesis (HLS) → HDL

Logic Synthesis → Technology Mapping → Packing → Placement → Routing → Timing (WS, TNS)

Runtime $t$
Leveraging Multi-Stage ML Inference

\[
RMSE = \sqrt{\frac{\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}{N}}
\]

\(\hat{y}_i\) = estimated timing
\(y_i\) = actual timing
\(N\) = number of samples

Runtime 0.4t

C/C++/OpenCL → High-Level Synthesis (HLS) → HDL

Logic Synthesis → Technology Mapping → Packing → Placement → Routing

ML Inference

Estimated Timing

RMSE = \(e\)
Leveraging Multi-Stage ML Inference

C/C++/OpenCL

High-Level Synthesis (HLS)

HDL

Logic Synthesis

Technology Mapping

Packing

Placement

Routing

Timing (WS, TNS)

ML Inference

Estimated Timing

RMSE = 1.2e

Runtime

0.07t
Leveraging Multi-Stage ML Inference

C/C++/OpenCL

High-Level Synthesis (HLS)

HDL

Logic Synthesis

Technology Mapping

Packing

Placement

Routing

Timing (WS, TNS)

ML Inference

Estimated Timing
RMSE = 1.6e

Trade-off between computing effort and estimation accuracy

C. Lo and P. Chow, “Multi-Fidelity Optimization for High-Level Synthesis Directives,” FPL’18
ML-Based Timing Estimation Framework

Design Stage	| Feature Type	| Design-Specific Features
---|---|---
Logic Synthesis, Technology Mapping, and Packing	| Resource	| #ALM, #LUT, #registers, #DSP, #I/O pins, #fan-out, etc.
| Timing	| WS, TNS

* XGBoost, [Chen et al., KDD’16]
LAMDA: Learning-Assisted Multi-Stage Design Autotuning

* OpenTuner [Ansel et al., PACT’14]
LAMDA – Evaluation I

### LAMDA – Evaluation I

**Online learning?**

<table>
<thead>
<tr>
<th>Yes</th>
<th>No</th>
</tr>
</thead>
<tbody>
<tr>
<td>online-multi (LAMDA)</td>
<td>offline-multi</td>
</tr>
<tr>
<td>online-single</td>
<td>offline-single</td>
</tr>
</tbody>
</table>

**Leveraging design-specific features from early stages?**

<table>
<thead>
<tr>
<th>Yes</th>
<th>No</th>
</tr>
</thead>
<tbody>
<tr>
<td>online-multi (LAMDA)</td>
<td>offline-multi</td>
</tr>
<tr>
<td>online-single</td>
<td>offline-single</td>
</tr>
</tbody>
</table>

**Design**

<table>
<thead>
<tr>
<th>Design</th>
<th>#ALUT</th>
<th>#FF</th>
<th>#DSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>dscg</td>
<td>6246</td>
<td>1679</td>
<td>4</td>
</tr>
</tbody>
</table>

**Global Best**

On average;

- 5.43x speedup compared to DATuner*
- 4.38x speedup compared to offline-single

* DATuner [Xu et al., FPGA’17]
LAMDA – Evaluation I

<table>
<thead>
<tr>
<th>Design</th>
<th>#ALUT</th>
<th>#FF</th>
<th>#DSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>dscg</td>
<td>6246</td>
<td>1679</td>
<td>4</td>
</tr>
</tbody>
</table>

Leveraging design-specific features from early stages?

<table>
<thead>
<tr>
<th>Online learning?</th>
<th>Yes</th>
<th>No</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>online-muti (LAMDA)</td>
<td>offline-muti</td>
</tr>
<tr>
<td></td>
<td>online-single</td>
<td>offline-single</td>
</tr>
</tbody>
</table>

Global Best

Global Mean

Design

#ALUT  #FF  #DSP
dscg 6246 1679 4
LAMDA – Evaluation II

Estimated versus actual QoR of design points visited
LAMDA: Learning-Assisted Multi-Stage Autotuning for FPGA Design Closure

Ecenur Ustun, Shaojie Xiang, Jinny Gui, Cunxi Yu, Zhiru Zhang

Electrical and Computer Engineering, Cornell University

Thank you!