Leveraging Prior Knowledge for Effective Design-Space Exploration in High-Level Synthesis

Authors:

Lorenzo Ferretti¹, Jihye Kwon², Giovanni Ansaloni¹, Giuseppe Di Guglielmo², Luca Carloni², Laura Pozzi¹,

¹ Università della Svizzera italiana, Lugano, Switzerland
² Columbia University, New York, United States

ESWEEK 2020
Motivation

Software description

```c
for(i;i<10;i++){
    A[i] = B[i]*i;
}
bar(&A,&B);
```

Loop unrolling

HLS

Hardware description

![Diagram showing Area vs Latency](chart.png)
for (i; i < 10; i++) {
    A[i] = B[i] * i;
} bar(&A, &B);

Motivation
Motivation

Application

Loop unrolling

Function inlining

Software description

for(i;i<10;i++){  
A[i] = B[i]*i;  
}  
bar(&A,&B);

Hardware description

Area

Latency
Table of Contents

✓ Motivation
  ➔ HLS-driven Design Space Exploration
  • State of the Art
  • The Idea: Leveraging Prior Knowledge
  • The Methodology
    • Signature Encoding
    • Similarity Evaluation
    • Inference Process
  • Results
HLS-driven Design Space Exploration (DSE)

Goal: get close to the Pareto solutions while minimising the number of synthesis.

Design space exploration problem

Legend
- Synthesised configurations
- Pareto configurations
- Exhaustive configurations
HLS-driven Design Space Exploration (DSE)

Goal: get close to the Pareto solutions while minimising the number of synthesis.
HLS-driven Design Space Exploration (DSE)

Goal: get close to the Pareto solutions while minimising the number of synthesis.
Table of Contents

✓ Motivation
✓ HLS-driven Design Space Exploration
➡ State of the Art for DSE
  • The Idea: Leveraging Prior Knowledge
  • The Methodology
    • Signature Encoding
    • Similarity Evaluation
    • Inference Process
  • Results
State of the Art for DSE

Two main approaches:

- Model-based methodologies

State of the Art for DSE

Two main approaches:
- Model-based methodologies
- Black-box-based methodologies
  - Training-based

State of the Art for DSE

Two main approaches:
- Model-based methodologies
- Black-box-based methodologies
  - Training-based
  - Refinement-based

State of the Art for DSE

Two main approaches:

- Model-based methodologies
- Black-box-based methodologies
  - Training-based
  - Refinement-based

---

Table of Contents

✓ Motivation
✓ HLS-driven Design Space Exploration
✓ State of the Art for DSE

➡ The Idea: Leveraging Prior Knowledge
  • The Methodology
    • Signature Encoding
    • Similarity Evaluation
    • Inference Process
  • Results
The Idea: Leveraging Prior Knowledge

Standard approach

$X_T$ Set of configurations explored in the DSE of a design T.
$P(T, X_T)$ Set of Pareto-optimal configurations identified with the DSE of T.
The Idea: Leveraging Prior Knowledge

Leveraging Prior Knowledge approach

$X_S$ Set of configurations explored in the DSE of a design S.
$P(S, X_S)$ Set of Pareto-optimal configurations identified with the DSE of S.

$X_T$ Set of configurations explored in the DSE of a design T.
$P(T, X_T)$ Set of Pareto-optimal configurations identified with the DSE of T.
Table of Contents

✓ Motivation
✓ HLS-driven Design Space Exploration
✓ State of the Art for DSE
✓ The Idea: Leveraging Prior Knowledge

➡ The Methodology
  • Signature Encoding
  • Similarity Evaluation
  • Inference Process
• Results
The Methodology

1. **Design & Config. Space**
   - Signature encoding
   - Target signature

2. **DSEs database**
   - Inference process
   - Source signature

3. **Target configurations**
   - HLS tool

4. **Synthesised designs**

   - HLS tool
   - Area - Latency
Table of Contents

✓ Motivation
✓ HLS-driven Design Space Exploration
✓ State of the Art for DSE
✓ The Idea: Leveraging Prior Knowledge
✓ The Methodology
  ➡ Signature Encoding
    • Similarity Evaluation
    • Inference Process
  • Results
The Methodology: Signature Encoding

A DSE is characterized with a simplified compact representation that abstracts the specification (code) and the associated configurations (set of applied directives).

**Signature encoding**: Specification Encoding & Configuration Space Descriptor

The **Specification Encoding (SE)** is a simplified representation of the original code describing those aspects of an HLS-application that can be targeted by HLS directives.

The **Configuration Space Descriptor (CSD)** describes the user-defined configuration space indicating the set of optimisations type and values considered for the DSE.
The Methodology: Signature Encoding

**Specification Encoding (SE):** simplified representation of the original code automatically generated through a compiler pass with LLVM.

Specification Encoding symbols:
- Functions —> \( F \)
- Function parameter passed by value —> \( V \)
- Function parameter passed by reference —> \( P \)
- Arrays definition or declaration —> \( A \)
- Structs definition or declaration —> \( S \)
- Loops —> \( L \)
- Load operations (e.g. \( a = \text{Array}[0] \)) —> \( R \)
- Store operations (e.g. \( \text{Array}[0] = a \)) —> \( W \)
- Function call —> \( \text{C#<function_name>} \)
- Scope —> \( \{ \} \)

```c
void last_step_scan(int bucket[BUCKETSIZE], int sum[SCAN_RADIX]) {
  int radixID, i, bucket_indx;
  last_1:
  for (radixID=0; radixID<SCAN_RADIX; radixID++) {
    last_2:
    for (i=0; i<SCAN_BLOCK; i++) {
      bucket_indx = radixID * SCAN_BLOCK + i;
      bucket[bucket_indx] = bucket[bucket_indx] + sum[radixID];
    }
  }
}
```

Running Example
The Methodology: Signature Encoding

**Configuration Space Descriptor (CSD):** a DSL is created to describe the optimisations type and values considered for the DSE. A CSD defines entirely the user-defined configuration space.

Example of CSD:
```
resource;last_step_scan;bucket;{RAM_2P_BRAM}
resource;last_step_scan;sum;{RAM_2P_BRAM}
array_partition;last_step_scan;bucket;1;
  {cyclic,block};{1->512,pow_2}
array_partition;last_step_scan;sum;1;
  {cyclic,block};{1->128,pow_2}
unroll;last_step_scan;last_1;{1->128,pow_2}
unroll last_step_scan;last_2 {1,2,4,8,16}
clock;{10}
```

```c
void last_step_scan(int bucket[BUCKETSIZE], int sum[SCAN_RADIX]) {
    int radixID, i, bucket_index;

    last_1: for (radixID=0; radixID<SCAN_RADIX; radixID++) {
        last_2: for (i=0; i<SCAN_BLOCK; i++) {
            bucket_index = radixID * SCAN_BLOCK + i;
            bucket[bucket_index] = bucket[bucket_index] + sum[radixID];
        }
    }
}
```
Table of Contents

✓ Motivation
✓ HLS-driven Design Space Exploration
✓ State of the Art for DSE
✓ The Idea: Leveraging Prior Knowledge
✓ The Methodology
  ✓ Signature Encoding
    ➡ Similarity Evaluation
      • Inference Process
      • Results
The Methodology: Similarity Evaluation

**Similarity Evaluation**: in order to identify the proper source for the inference process the similarity among a target DSE and the available source is calculated. The similarity function (\( \text{Sim} \)) is given by a linear combination of Signature Encoding similarity (\( \text{Sim}_{SE} \)) and Configuration Space Descriptor similarity (\( \text{Sim}_{CSD} \)).

\[
\text{Sim} = \alpha \text{Sim}_{SE} + (1 - \alpha) \text{Sim}_{CSD} \quad \alpha \in [0,1]
\]

\( \text{Sim}_{SE} = \text{LCS}(SE_T, SE_S) \)

\[
\text{Sim}_{CSD} = 1 - \left[ \frac{1}{I} \sum_{i=1}^{I} \Delta(K_i, M_{T,S}(K_i))/D_{\text{MAX}} \right]
\]

\[
\Delta(K_i, K_j) = \sqrt{\sum_{n=1}^{|K_i|} \sum_{m=1}^{|K_j|} (\min_{k} |\delta(k_n, k_m)|)^2} \quad k_n \in K_i, k_m \in K_j
\]

\[
\delta(k_n, k_m) = \sqrt{\sum_{z=1}^{Z} |k_{n,z}, k_{m,z}|^2}
\]
The Methodology: Similarity Evaluation

**CSD similarity**: measures the similarity among knobs of target and source CSDs.

A top-down mapping maps knobs of the source Signature Encoding to knobs of the target one.

---

**Running Example**

<table>
<thead>
<tr>
<th>CSD_{Source}</th>
<th>SE_{Source}</th>
<th>CSD_{Target}</th>
<th>SE_{Target}</th>
</tr>
</thead>
<tbody>
<tr>
<td>array_partition; get_delta_matrix_weights2; delta_weights2; 1; {cyclic, block}; {1-&gt;256, pow_2}</td>
<td></td>
<td>resource; last_step_scan; bucket; {RAM_2P_BRAM}</td>
<td></td>
</tr>
<tr>
<td>array_partition; get_delta_matrix_weights2; output_difference; 1; {cyclic, block}; {1-&gt;64, pow_2}</td>
<td></td>
<td>array_partition; last_step_scan; bucket; 1; {cyclic, block}; {1-&gt;512, pow_2}</td>
<td></td>
</tr>
<tr>
<td>array_partition; get_delta_matrix_weights2; last_activations; 1; {cyclic, block}; {1-&gt;64, pow_2}</td>
<td></td>
<td>array_partition; last_step_scan; sum; {RAM_2P_BRAM}</td>
<td></td>
</tr>
<tr>
<td>unroll; get_delta_matrix_weights2; loop_1; {1-&gt;64, pow_2}</td>
<td></td>
<td>unroll; last_step_scan; last_1; {1-&gt;128, pow_2}</td>
<td></td>
</tr>
<tr>
<td>unroll; get_delta_matrix_weights2; loop_2; {1-&gt;64, pow_2}</td>
<td></td>
<td>unroll; last_step_scan; last_2; {1, 2, 4, 8, 16}</td>
<td></td>
</tr>
<tr>
<td>clock; {10}</td>
<td></td>
<td>clock; {10}</td>
<td></td>
</tr>
</tbody>
</table>
The Methodology: Similarity Evaluation

**CSD similarity**: measures the similarity among knobs of target and source CSDs.

<table>
<thead>
<tr>
<th>Domain</th>
<th>Mapping</th>
<th>Domain</th>
<th>Knob values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target knobs</td>
<td>$K_1$ Resource</td>
<td>Source knobs</td>
<td>$K_2$ Resource</td>
</tr>
<tr>
<td></td>
<td>$K_3$ Part. type Part. factor</td>
<td></td>
<td>$K_4$ Part. type Part. factor</td>
</tr>
<tr>
<td></td>
<td>$K_5$ Unroll</td>
<td></td>
<td>$K_6$ Unroll</td>
</tr>
<tr>
<td></td>
<td>$K_7$ Clock</td>
<td></td>
<td>$K_6$ Clock</td>
</tr>
<tr>
<td></td>
<td>$K_1$ Part. type Part. factor</td>
<td></td>
<td>$K_2$ Part. type Part. factor</td>
</tr>
<tr>
<td></td>
<td>$K_3$ Part. type Part. factor</td>
<td></td>
<td>$K_4$ Part. type Part. factor</td>
</tr>
<tr>
<td></td>
<td>$K_5$ Unroll</td>
<td></td>
<td>$K_6$ Unroll</td>
</tr>
<tr>
<td></td>
<td>$K_7$ Clock</td>
<td></td>
<td>$K_6$ Clock</td>
</tr>
</tbody>
</table>

Set of values for target knob $K_6$:
- 1
- 2
- 4
- 6
- 8
- 16
- 32
- 64
- 128

Set of values for source knob $K_5$:
- 1
- 2
- 4
- 6
- 8
- 16
- 32
- 64
The Methodology: Similarity Evaluation

$Sim_{SE}$

$Sim_{CSD}$
The Methodology: Similarity Evaluation

Encoding Signature similarity

Source ID

Target ID
<table>
<thead>
<tr>
<th>Topic</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motivation</td>
<td>✓</td>
</tr>
<tr>
<td>HLS-driven Design Space Exploration</td>
<td>✓</td>
</tr>
<tr>
<td>State of the Art for DSE</td>
<td>✓</td>
</tr>
<tr>
<td>The Idea: Leveraging Prior Knowledge</td>
<td>✓</td>
</tr>
<tr>
<td>The Methodology</td>
<td>✓</td>
</tr>
<tr>
<td>Signature Encoding</td>
<td>✓</td>
</tr>
<tr>
<td>Similarity Evaluation</td>
<td>✓</td>
</tr>
<tr>
<td>Inference Process</td>
<td>➡</td>
</tr>
<tr>
<td>Results</td>
<td>•</td>
</tr>
</tbody>
</table>

**Table of Contents**
The methodology: Inference Process

\( x_S = [x_S^1, \ldots, x_S^j] \in X_S \quad x_T = [x_T^1, \ldots, x_T^j] \in X_T \)

\( x_T^i = \arg\min_n \{\delta(k_n, x_S^j)\} \)

\( \delta(k_n, k_m) = \sqrt{\sum_{z=1}^{Z} |k_{n,z}, k_{m,z}|^2} \)
The methodology: Inference Process

Running Example
The methodology: Inference Process

The source design space is iteratively peeled and lower-rank Pareto frontiers are used for the inference.
The methodology: Inference Process

The source design space is iteratively peeled and lower-rank Pareto frontiers are used for the inference.
The methodology: Inference Process

The source design space is iteratively peeled and lower-rank Pareto frontiers are used for the inference.
Table of Contents

✓ Motivation
✓ HLS-driven Design Space Exploration
✓ State of the Art for DSE
✓ The Idea: Leveraging Prior Knowledge
✓ The Methodology
  ✓ Signature Encoding
  ✓ Similarity Evaluation
  ✓ Inference Process
➡ Results
Results

We have considered 39 out of 50 possible design from Machsuite[4].

For each of them we have performed an exhaustive exploration and used it as a ground-truth to evaluate the quality of the DSE.

We have used Average Distance from Reference Set (ADRS) metric to measure the distance among the retrieved Pareto frontier and the ground-truth.

\[
ADRS(\bar{P}, P) = \left[ \frac{1}{|P|} \sum_{p \in P} \min_{\bar{p} \in \bar{P}} (d(\bar{p}, p)) \right]
\]

\[
d(\bar{p}, p) = \max\{0, (A_{\bar{p}} - A_p)/A_p, (L_{\bar{p}} - L_p)/L_p\}
\]

Results

Effectiveness of the similarity metric: 1st-ranked source.

✓ High Specification Encoding similarity
✓ High Configuration Space Descriptor similarity
Results

Effectiveness of the similarity metric: 30th-ranked source.

ADR$S=0.85$

- \( \times \) Low Specification Encoding similarity
- \( \checkmark \) High Configuration Space Descriptor similarity
Results

Effectiveness of the similarity metric: 35th-ranked source.

✓ High Specification Encoding similarity

✗ Low Configuration Space Descriptor similarity
Effectiveness of the similarity metric: 37th-ranked source.

Results

- Low Specification Encoding similarity
- Low Configuration Space Descriptor similarity
Effectiveness of the similarity metric: source ranking.
Results

Effectiveness of the similarity metric: selection criterion.
Results

Effectiveness of the similarity metric: influence of multiple Pareto frontier rank inference.
Comparison of our methodology with respect to SoA ones for similar problems size.

| ||CS|| | Prior Knowl. | Lattice | Cluster | RF-TED | Zhong |
|---|---|---|---|---|---|---|
| < 200 | 7 | 36 | 37 | 155 | NA |
| < 700 | 10 | 64 | 64 | 391 | 19 |
| < 1800 | 22 | 230 | 290 | 1588 | 31 |
| < 6000 | 19 | 460 | 460 | 1903 | 32 |
| < 16000 | NA | NA | NA | NA | 35 |
| < 32000 | 38 | NA | NA | NA | NA |

Number of synthesis required to reach an ADRS goal of 0.04.

Table of Contents

✓ Motivation
✓ HLS-driven Design Space Exploration
✓ State of the Art for DSE
✓ The Idea: Leveraging Prior Knowledge
✓ The Methodology
  ✓ Signature Encoding
  ✓ Similarity Evaluation
  ✓ Inference Process
✓ Results

Database of DSEs will be released soon!

Thank you for your attention!