On Channel Restructuring for Complete FIFO Recovery
Christophe Alias

To cite this version:
Christophe Alias. On Channel Restructuring for Complete FIFO Recovery. ICCD 2019 - 37th IEEE International Conference on Computer Design, Nov 2019, Abu Dhabi, United Arab Emirates. hal-02433318

HAL Id: hal-02433318
https://inria.hal.science/hal-02433318
Submitted on 9 Jan 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
On Channel Restructuring for Complete FIFO Recovery

Focus: Dataflow models as an intermediate representation for High-Level Synthesis
Challenge: Recover FIFO channels after code restructuring
Contributions: HLS algorithm for channel restructuring + Dataflow model to ensure completeness

1) Polyhedral Process Networks

\[
\begin{align*}
\text{for } i & = 0 \text{ to } N + 1; \\
\text{Load}(i); \\
\text{for } t & = 1 \text{ to } T; \\
\text{for } i & = 1 \text{ to } N; \\
\text{store}(i); \\
\text{for } i & = 1 \text{ to } N; \\
\text{store}(i); \\
\end{align*}
\]

2) Data-aware Process Networks

\[
\begin{align*}
\text{for } i & = 0 \text{ to } N + 1; \\
\text{Load}(i); \\
\text{for } t & = 1 \text{ to } T; \\
\text{for } i & = 1 \text{ to } N; \\
\text{store}(i); \\
\text{for } i & = 1 \text{ to } N; \\
\text{store}(i); \\
\end{align*}
\]

3) Our FIFO Recovery Algorithm

\[
\begin{align*}
\text{for } i & = 0 \text{ to } N + 1; \\
\text{Load}(i); \\
\text{for } t & = 1 \text{ to } T; \\
\text{for } i & = 1 \text{ to } N; \\
\text{store}(i); \\
\text{for } i & = 1 \text{ to } N; \\
\text{store}(i); \\
\end{align*}
\]

Theorem: FIFO recovery is complete on our DPN model

4) Experimental Results

<table>
<thead>
<tr>
<th>Kernel</th>
<th>DPN</th>
<th>PPN with tiling</th>
<th>DPN</th>
<th>PPN with tiling</th>
</tr>
</thead>
<tbody>
<tr>
<td>trisolv</td>
<td>1</td>
<td>100</td>
<td>1</td>
<td>100</td>
</tr>
<tr>
<td>cholesky</td>
<td>2</td>
<td>100</td>
<td>2</td>
<td>100</td>
</tr>
<tr>
<td>lu</td>
<td>3</td>
<td>100</td>
<td>3</td>
<td>100</td>
</tr>
<tr>
<td>symm</td>
<td>4</td>
<td>100</td>
<td>4</td>
<td>100</td>
</tr>
<tr>
<td>trmm</td>
<td>5</td>
<td>100</td>
<td>5</td>
<td>100</td>
</tr>
<tr>
<td>trisolv</td>
<td>6</td>
<td>100</td>
<td>6</td>
<td>100</td>
</tr>
<tr>
<td>cholesky</td>
<td>7</td>
<td>100</td>
<td>7</td>
<td>100</td>
</tr>
<tr>
<td>lu</td>
<td>8</td>
<td>100</td>
<td>8</td>
<td>100</td>
</tr>
<tr>
<td>symm</td>
<td>9</td>
<td>100</td>
<td>9</td>
<td>100</td>
</tr>
<tr>
<td>trmm</td>
<td>10</td>
<td>100</td>
<td>10</td>
<td>100</td>
</tr>
<tr>
<td>trisolv</td>
<td>11</td>
<td>100</td>
<td>11</td>
<td>100</td>
</tr>
<tr>
<td>cholesky</td>
<td>12</td>
<td>100</td>
<td>12</td>
<td>100</td>
</tr>
</tbody>
</table>

Experimental setup:
- We have run our algorithm on the kernels of PolyBench/C v3.2
- We have checked the completeness of our algorithm on PPN with DPN partitioning (DPN)
- We have studied the behavior of our algorithm on general PPN, without DPN partitioning (PPN with tiling).

Christophe Alias, Faculty Research Scientist
INRIA, ENS de Lyon, France

http://perso.ens-lyon.fr/christophealias