Domain-Specific Augmentations for High-Level Synthesis

ASAP’14, 19 June 2014

M. Schmid, A. Tanase, V. Badhouria, F. Hannig, J. Teich and D. Ghoshal
Domain-Specific Augmentations for HLS

- Huge progress over the past years
- Tools must provide a high degree of flexibility
- Competitive solutions require deep domain-specific knowledge
- Most tools optimized with respect to specific domain (e.g. image and signal processing)
- Different areas of expertise
  - Which algorithm is suitable?
  - How to parallelize the algorithm?
  - How can the algorithm be implemented?
- Domain-specific augmentations aim at combining specific knowledge and make it available at a high abstraction level
Example: Noise Reduction
Example: Noise Reduction

● Which algorithm to use?
  ● Image contains salt and pepper noise, which can be effectively reduced by a median filter.

● How to implement?
  ● Sorting network (FPGA, ASIC)
  ● Systolic sorter (processor array)
  ● Sorting algorithms, such as merge or quick sort (CPU)

● How to parallelize?
  ● Pipelining
  ● Partitioning
Design Flow

Program in the PAULA Language

PARO

Mathematical Libraries and Solvers → HIGH-LEVEL TRANSFORMATIONS

ARCHITECTURE MODEL

HIGH-LEVEL TRANSFORMATIONS → POLYHEDRAL TRANSFORMATIONS

Simulation (PARO) → Test Bench Generation

HARDWARE SYNTHESIS

Hardware Description (VHDL) → Simulation (External Simulator)
Image Processing in PARO

- Image processing can be classified into point, local, and global operators.
- Local operators (filtering an image with a kernel) use a region of the input image (typically a window) to derive a single pixel in the output image.
Local Operators in PARO

- Problem: Missing pixels at the image borders
- One solution is to simply ignore these cases
- Example: Averaging filter using the SUM reduction operator:

\[
\begin{align*}
\text{PAR}(i \geq 0 \text{ and } i < M \text{ and } j \geq 0 \text{ and } j < N) & \{
\text{x\_out}[i,j] = \text{SUM}[l \geq -1 \text{ and } l \leq 1 \text{ and } m \geq -1 \text{ and } m \leq 1](x[i+l,j+m])/9 \\
& \text{if}(i \geq 1 \text{ and } i < M-1 \text{ and } j \geq 1 \text{ and } j < N-1);
\text{x\_out}[i,j] = x[i,j] \\
& \text{if}(i < 1 \text{ or } i \geq M-1 \text{ or } j < 1 \text{ or } j \geq N-1);
\}
\end{align*}
\]

- Leaves strip of unprocessed pixels around the image
- More sophisticated solution: Border treatment
Border Treatment in PAULA

In a functional language, such as PAULA, border treatment requires additional statements for how to replace missing pixels.

For example, a small 3x3 window filter requires 81 statements.

- 9 different cases
- 9 statements per case

```par
PAR(i>=0 and i<M and j>=0 and j<N)
{ // inner image (1)
    x_0[i,j]=x[i-1,j-1]
    if(i>0 and i<M and j>0 and j<N);

    // corner cases (2-5)
    x_0[i,j]=x[i+1,j+1] if(i==0 and j==0);
    x_0[i,j]=x[i+1,j-1] if(i==0 and j==N-1);
    x_0[i,j]=x[i-1,j+1] if(i==M-1 and j==0);
    x_0[i,j]=x[i-1,j-1] if(i==M-1 and j==N-1);

    // border cases (6-9)
    x_0[i,j]=x[i+1,j-1] if(i==0 and j>0 and j<N);
    x_0[i,j]=x[i-1,j+1] if(i==0 and j>0 and j<N-1);
    x_0[i,j]=x[i-1,j-1] if(i==M-1 and j>0 and j<N-1);
    x_0[i,j]=x[i-1,j-1] if(i==M-1 and j>0 and j<N-1);
}
```
Domain-Specific Augmentation: Border Treatment

- Instead, we propose to make different border treatment schemes available as a high-level transformation in PARO

`#pragma PARO_BT type=MIRROR_101 var=x`

- Example: Apply Mirror 101 border treatment to input variable x
Sorting and Median Computation

- Sorting and median computation requires a special implementation in a functional language.
Domain-Specific Augmentation: Sort and Med

- We propose sorting and median computation as domain-specific augmentations in the form of reduction operators in PAULA

  //sorting
  SORT[iteration_space](expression);

  //median computation
  MED[iteration_space](expression)

- A median filter can then be specified as

  #pragma PARO_BT type=MIRROR_101 var=x

  PAR(i>=0 and i<M and j>=0 and j<N){
    y[i,j] = MED[k>=-1 and k<=1 and l>=-1 and l<=1]
             (x[i+k,j+l]);
  }
Thank you for the attention!

Domain-Specific Augmentations for High-Level Synthesis

M. Schmid, A. Tanase, F. Hannig, and J. Teich
Hardware/Software Co-Design
University of Erlangen-Nürnberg
Erlangen, Germany
moritz.schmid@fau.de

V. Bhadouria and D. Ghoshal
Department of Electronics and Communication Engineering
National Institute of Technology
Agartala, India