Global optimization of folding, padding and FIFO sizing in FINN

Lukas Stasytis[1,2], Thomas Preußer[1], Jakoba Petri-Koenig[1]

AMD[1], TU-Darmstadt[2]

Motivation

Before handing off a FINN-ONNX model to the FPGA toolchain, we perform:

1. **Folding**: set the degree of parallelism for each node, done manually and relies on good understanding of FINN (error-prone). Features strict restrictions on possible values we can pick
2. **FIFO sizing:** set the buffer depths between nodes, done automatically using RTL simulation, making it very time-consuming (synthesis times)

- These problems are all interlinked
- We can solve them in a unified manner

Contributions

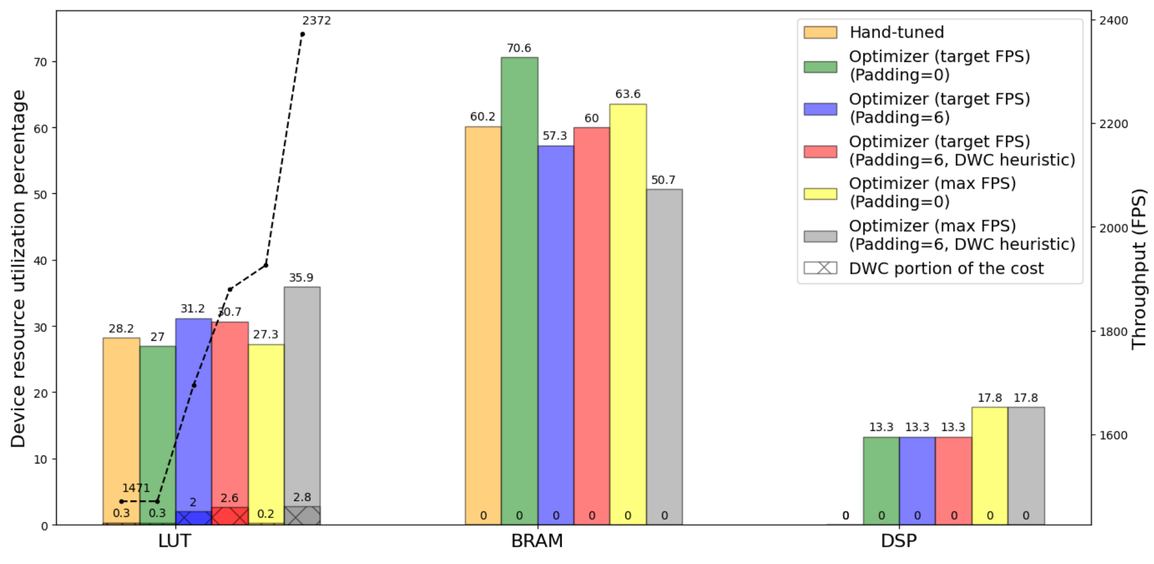
1. Global minimizer-based **SetFolding()** transformation which sets the folding factors of all nodes in a FINN-ONNX model, considering folding effects on:
   - FIFO sizes
   - Total model resource consumption & throughput
   - Data Width Converter (**DWC**) insertions
   - Potential padding of nodes

2. Generalized DWC which allows arbitrary padding and cropping of input streams to enable higher degrees of parallelism in FINN nodes & framework for padding each node

3. Analytical characteristic functions for FINN-HLSLIB nodes which allow automatic FIFO sizing without needing to run RTLSIM

Results

Model: **CNV-w1a1**, final bitstream resource consumption and RTLSIM throughput (as FPS). Comparing hand-tuned and optimized folding.



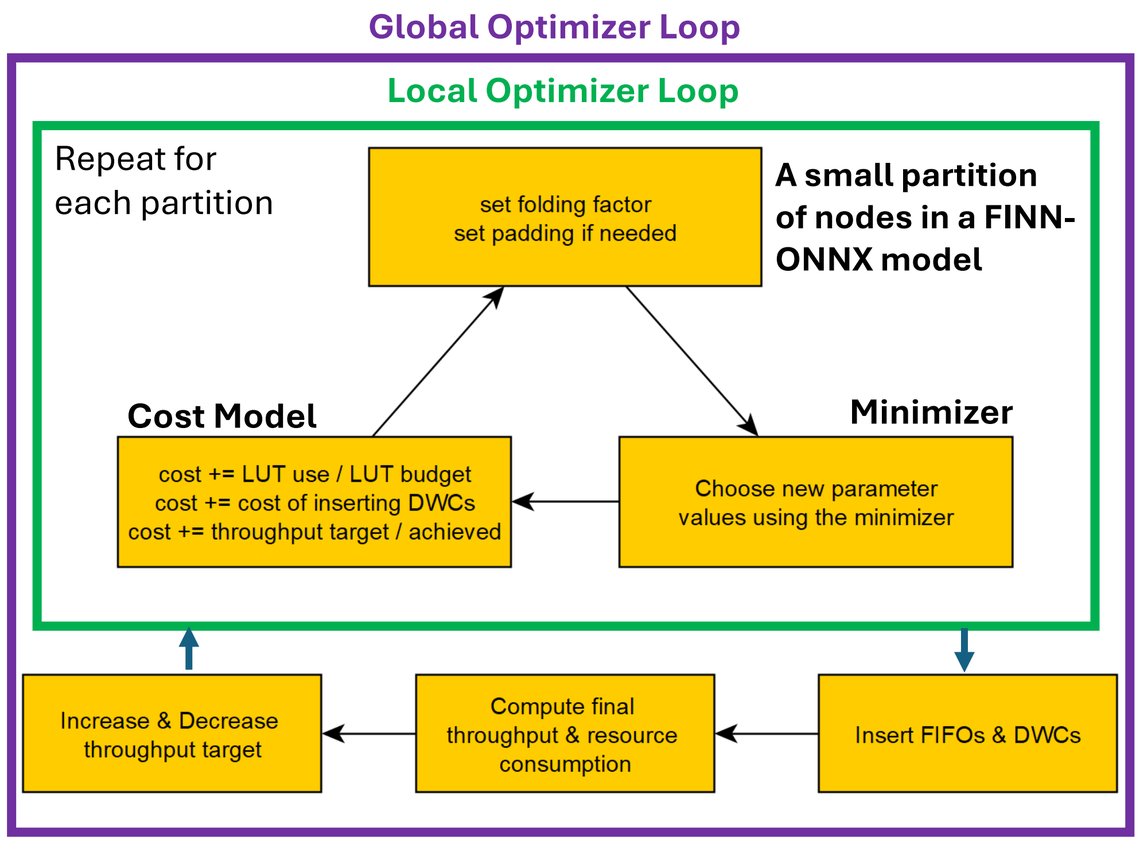
- Folding optimizer compile time: **seconds** to **minutes**
- FIFO sizing time: **1-30 seconds / node, 100x faster than RTLSIM**
- Tested on all models in ***finn-examples*** except **resnet50**
- Optimizer PR to ***finn/dev***, padding support to ***finn/experimental***
- HLS General DWC variant >**10x less** resource efficient than RTL variant of old DWC, will be converted to RTL-only eventually
- FIFO sizing is NOT accurate for branching path models yet! (But is a decent approximation as to the effects of folding)

These additions now enable design space exploration in terms of folding

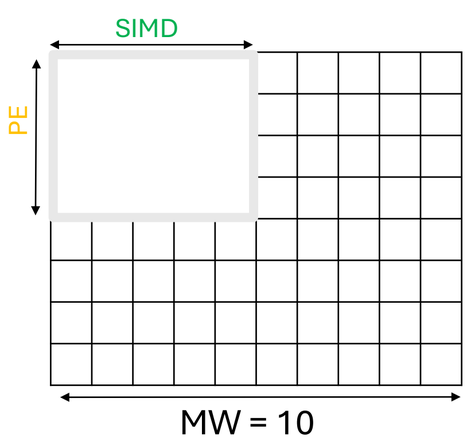
Key Ideas

1. Folding Optimizer:

- Treat the task as a global bounded-integer optimization problem
- Minimizing with simulated annealing **(scipy.minimize.dual_basin)**
- Manage the curse of dimensionality using divide and conquer
- New heuristics are introduced by updating the cost model
- Analytical FIFO sizing allows putting it in the loop as well
- Padding is considered to increase search space

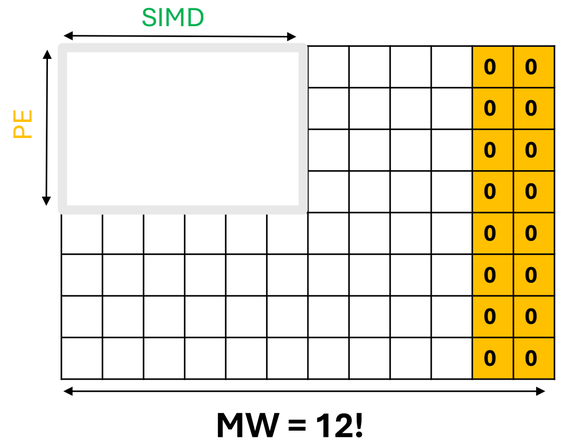


2. Padding & DWC:



Possible SIMD choices: [1,2,5,10]

What if I want SIMD = 6?

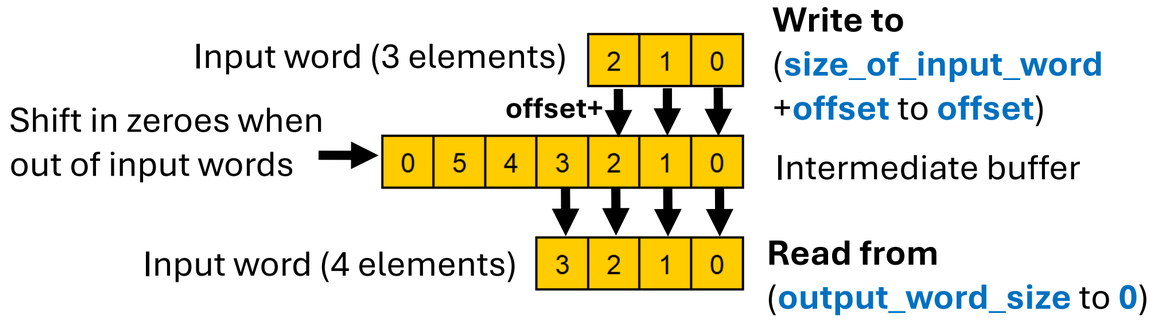


Possible SIMD choices: [1,2,3,4,6,12]

[1,2,3,4,5,6,10,12] possible factor super set! (apply padding based on picked folding factor)

How to introduce the zeroes? The DataWidthConverter!

Redesigned to a shift-register structure with dynamic write addressing to maintain state (we track how many elements are left in the buffer after a write). Padding and cropping introduced by tracking read and written words and either ending stream writes early or shifting in zeroes.



Contact: lukas.stasytis@tu-darmstadt.de