fpgaConvNet: Mapping Convolutional Neural Networks on Embedded FPGAs

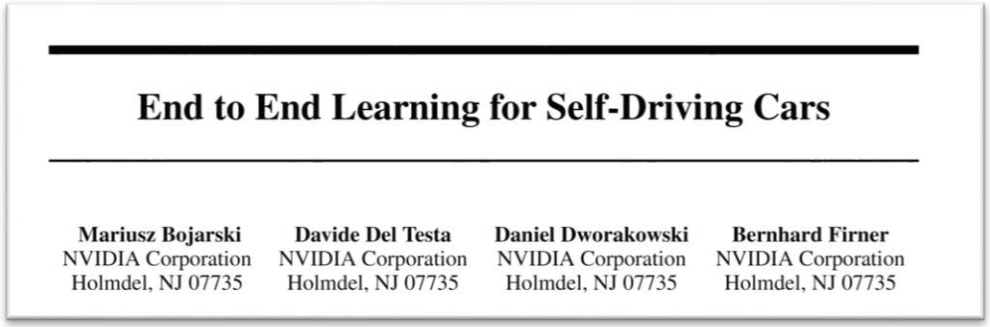
Stylianos I. Venieris and Christos-Savvas Bouganis

sv1310@ic.ac.uk

Electrical and Electronic Engineering Department

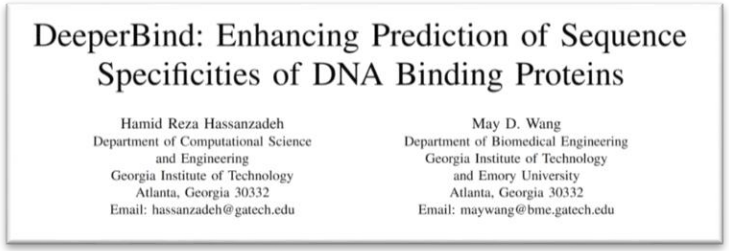
Imperial College London

Deep Learning-enabled AI Applications



End to End Learning for Self-Driving Cars

Mariusz Bojarski NVIDIA Corporation Holmdel, NJ 07735
Davide Del Testa NVIDIA Corporation Holmdel, NJ 07735
Daniel Dworakowski NVIDIA Corporation Holmdel, NJ 07735
Bernhard Firner NVIDIA Corporation Holmdel, NJ 07735



DeeperBind: Enhancing Prediction of Sequence Specificities of DNA Binding Proteins

Hamid Reza Hassanzadeh
Department of Computational Science and Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332
Email: hassanzadeh@gatech.edu

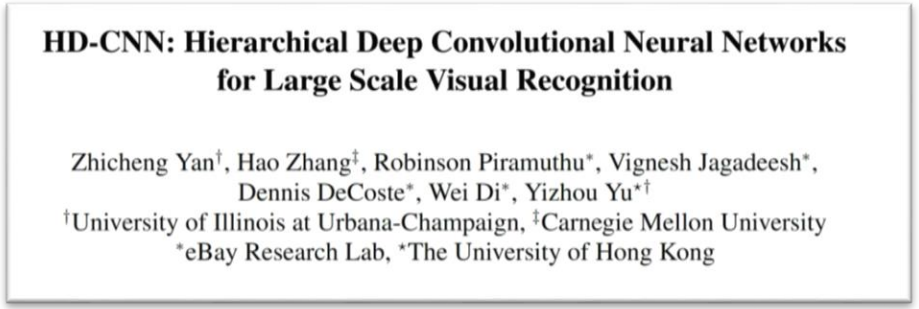
May D. Wang
Department of Biomedical Engineering
Georgia Institute of Technology and Emory University
Atlanta, Georgia 30332
Email: maywang@bme.gatech.edu

Learning visual similarity for product design with convolutional neural networks

Sean Bell Kavita Bala
Cornell University*

Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks

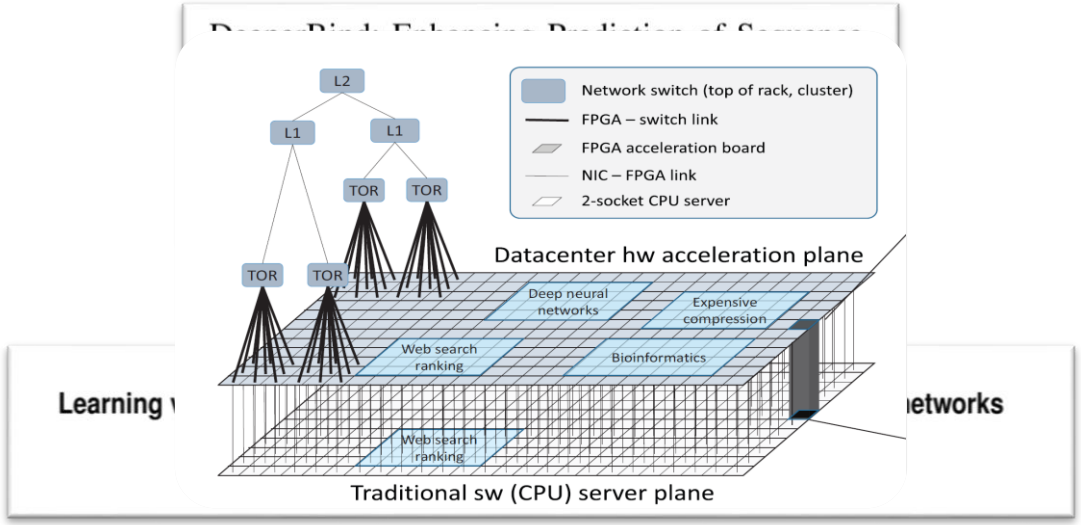
Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun



HD-CNN: Hierarchical Deep Convolutional Neural Networks for Large Scale Visual Recognition

Zhicheng Yan†, Hao Zhang‡, Robinson Piramuthu*, Vignesh Jagadeesh*, Dennis DeCoste*, Wei Di*, Yizhou Yu*†
†University of Illinois at Urbana-Champaign, ‡Carnegie Mellon University
*eBay Research Lab, *The University of Hong Kong

Deep Learning-enabled AI Applications



Application-level Performance Requirements for Neural Networks

Image Captioning (Google, 2016)

A brown bear standing on top of a lush green field.

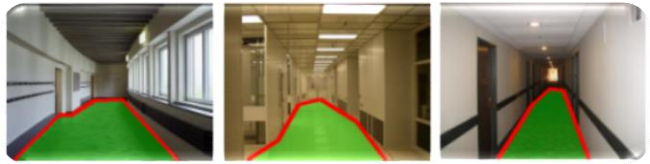
Uploaded Image

Video Surveillance

Search Results

Visual Search (Baidu, 2015)

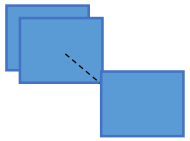
Autonomous Driving (NVIDIA, 2016)



Scene Understanding (CMU, 2016)

High-Throughput Applications

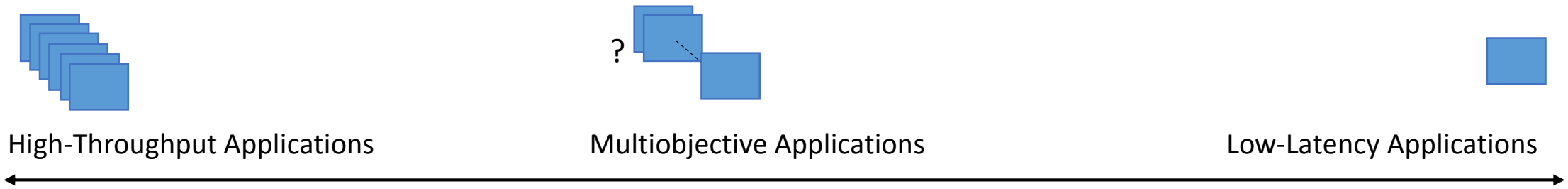
?



Multiobjective Applications

Low-Latency Applications

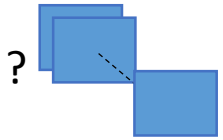
Application-level Performance Requirements for Neural Networks



Power constraints

- Absolute power consumption
- Performance-per-Watt

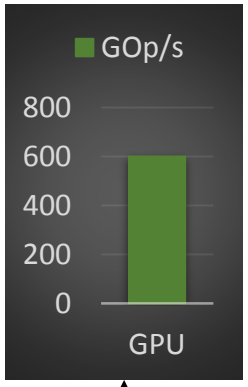
Embedded Platforms for Neural Networks



High-Throughput Applications

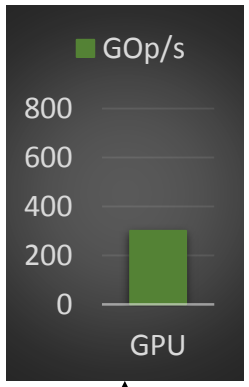
Multiobjective Applications

Low-Latency Applications

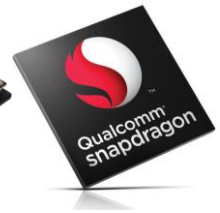


Favourable batch size

Batch size of 1



GPUs – Tegra K1, X1 and X2
DSPs – Qualcomm Hexagon,
Apple Neural Engine, …



Power constraints

- Absolute power consumption
- Performance-per-Watt

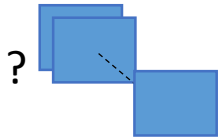
FPGAs

FPGAs

- Heterogeneous resources
  - Coarse compute units (DSPs)
  - Logic gates and storage elements
  - On-chip memory
- Programmable interconnections

- *Customisation*
  - Custom datapaths
  - Custom memory subsystems
- *Reconfigurability*

FPGAs for Neural Networks



High-Throughput Applications

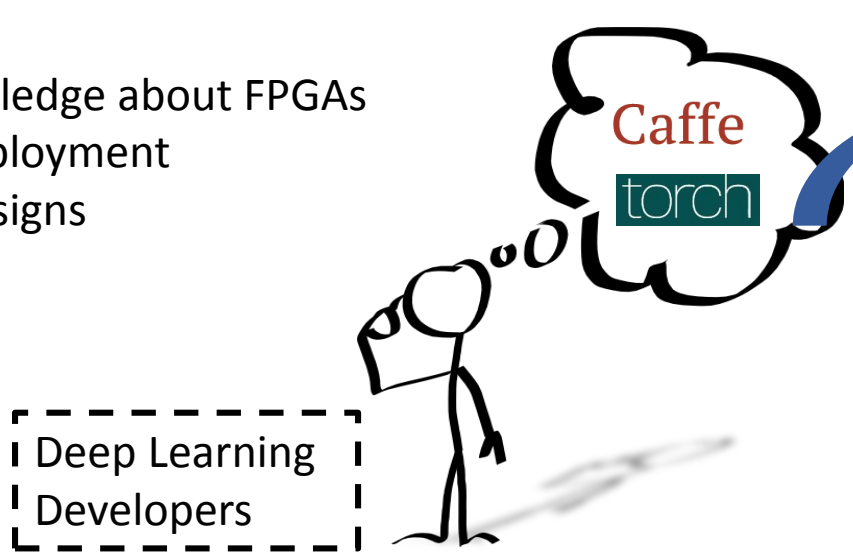
Multiobjective Applications

Low-Latency Applications

Power constraints

- Absolute power consumption
- Performance-per-Watt

Little knowledge about FPGAs
Ease of deployment
"Good" designs

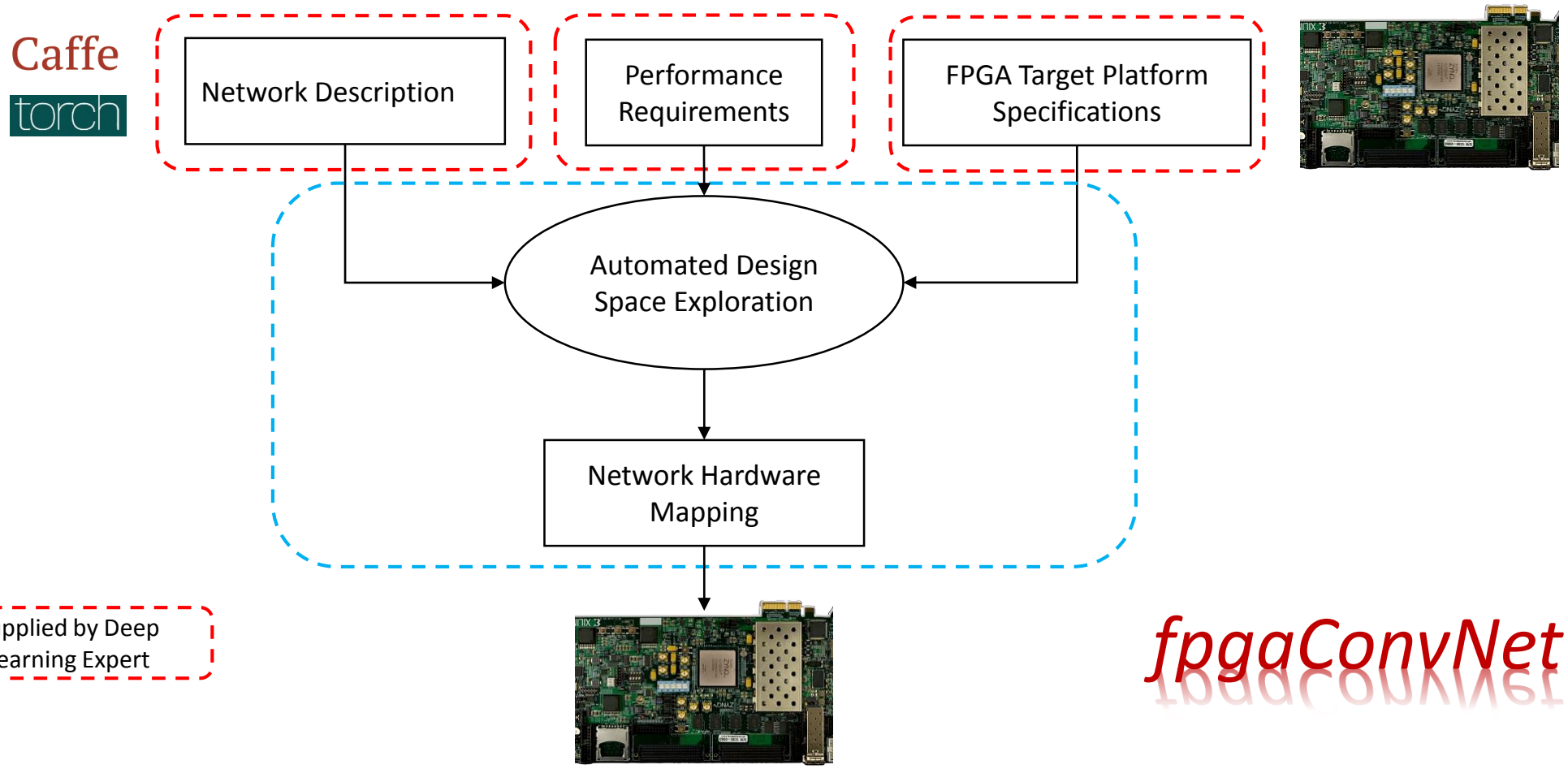


?

Challenges:

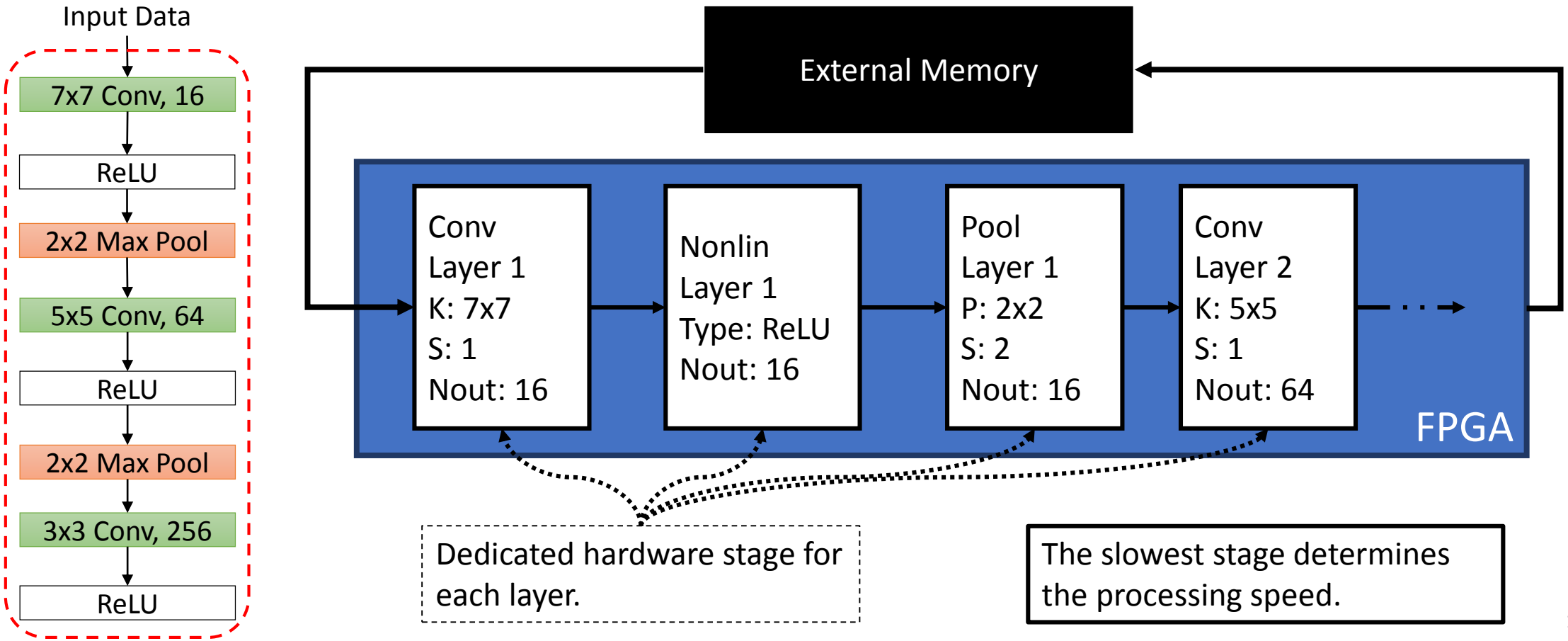
- High-dimensional design space
- Diverse application-level needs
- Utilise the FPGA resources
- Design automation

Automated CNN-to-FPGA Design Flow



Streaming Architecture for FPGA-based CNNs

- GPU-based CNNs are restricted to *sequential layer-by-layer* execution
- FPGA-based CNNs can *overlap* the execution of layers



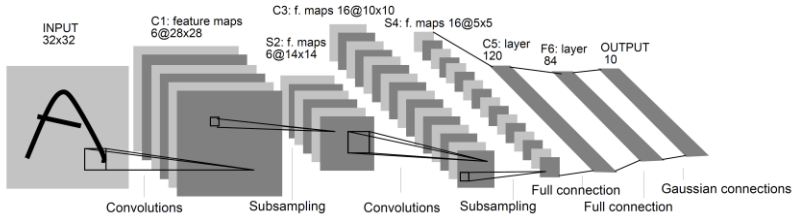
fpgaConvNet – Key Characteristics

- Differentiating factors
  - Streaming architecture
  - Hardware design tailored to the target (CNN, FPGA) pair
  - No limit on #weights and model size

- A Synchronous Dataflow model for CNNs
  - CNN as a data-driven graph
  - Workload is represented as a matrix
  - Each layer is mapped to a tunable set of hardware building blocks

- Design space exploration based on **transformations**
  - Coarse-grained folding
  - Fine-grained folding
  - Graph partitioning with reconfiguration
  - Weights reloading

Streaming

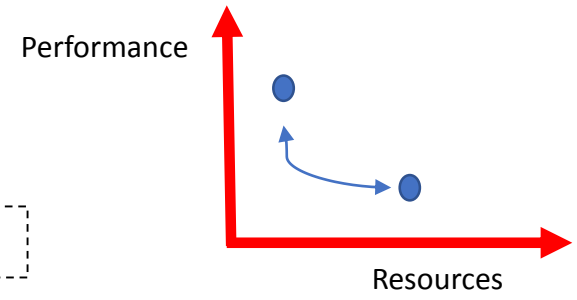


Analytical Power

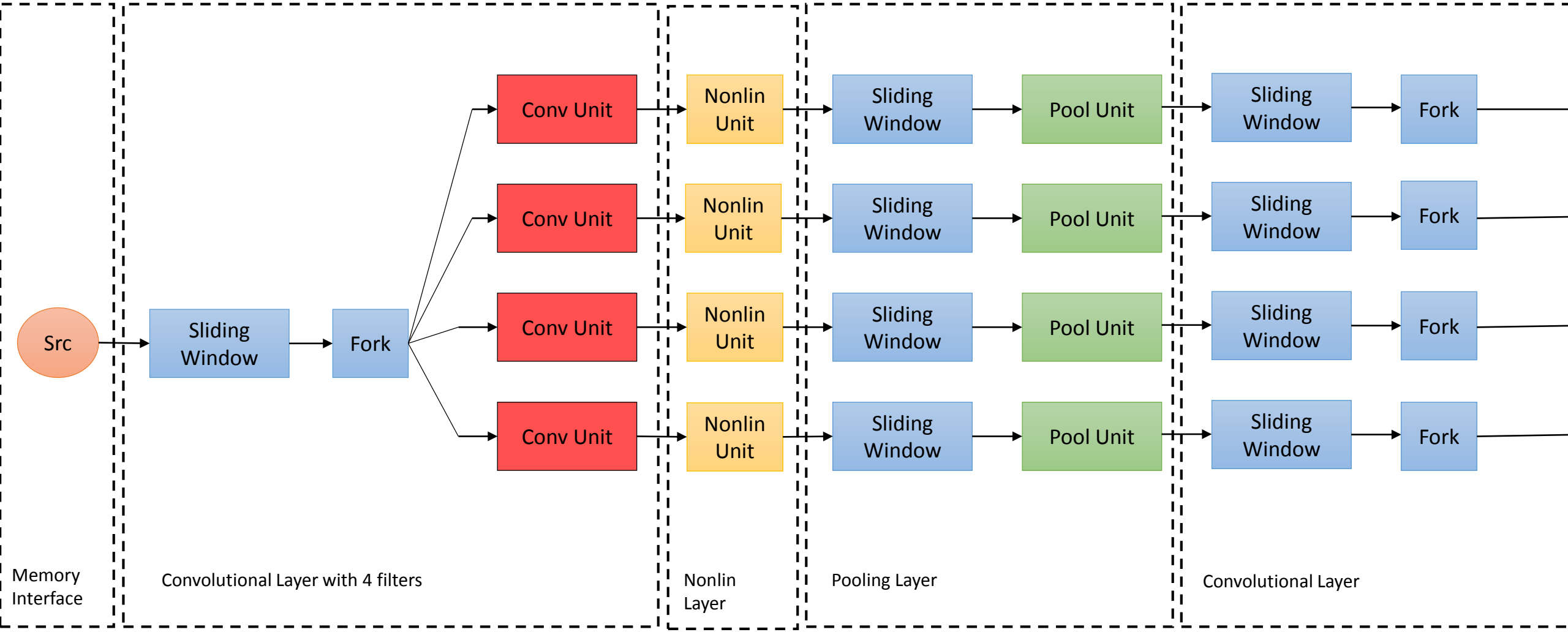
Max Throughput or Min Latency

$$t_{total}(B, N_P, \mathbf{\Gamma}) = \sum_{i=1}^{N_P} t_i(B, \mathbf{\Gamma}_i) + (N_P - 1) \cdot t_{reconfig.}$$

Customisation

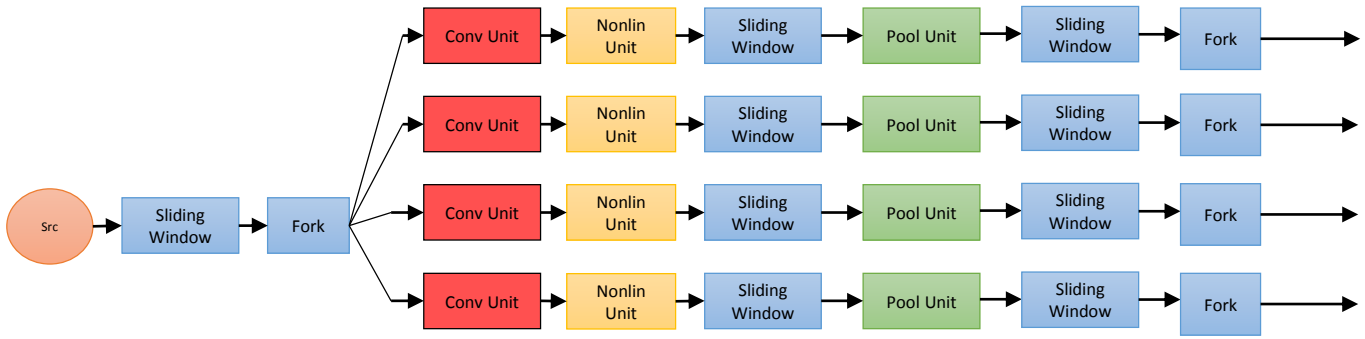


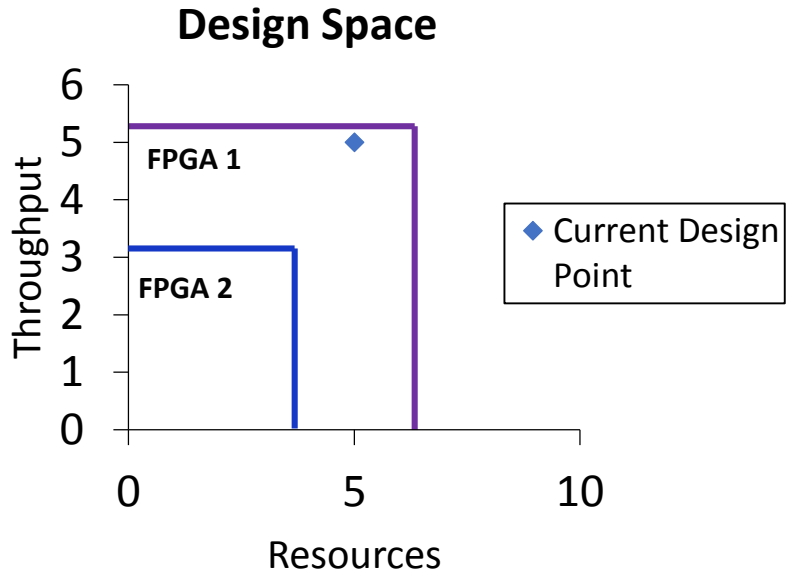
fpgaConvNet – Streaming Architecture for CNNs



fpgaConvNet – Streaming Architecture for CNNs

CNN Hardware SDF Graph



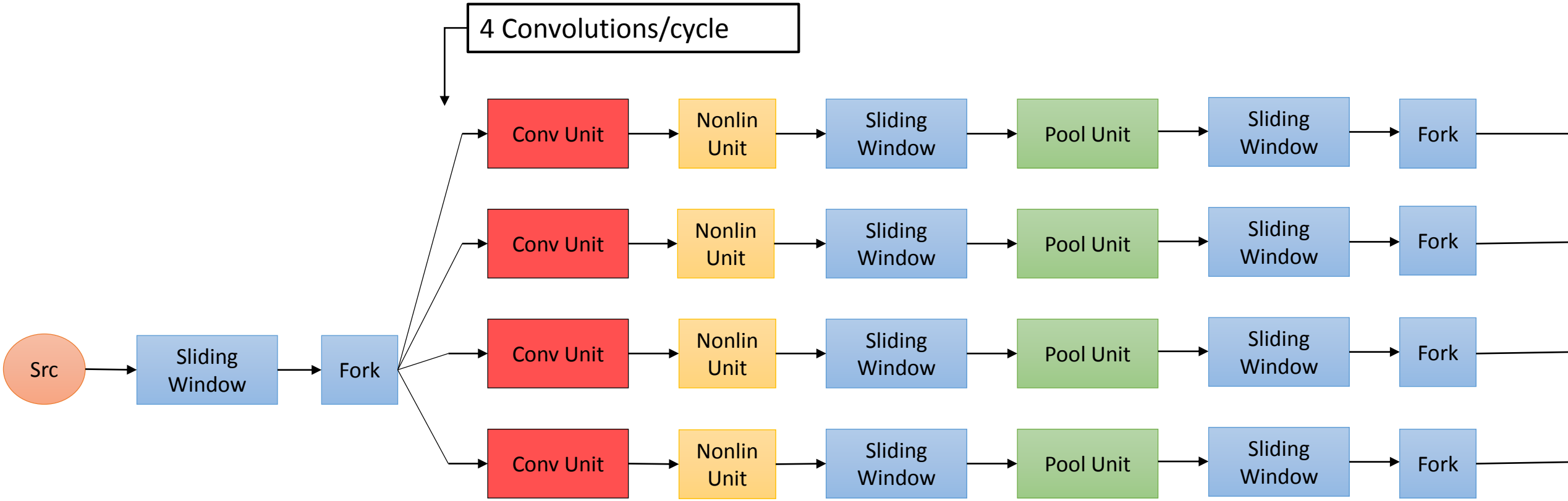


Complex Model → Bottlenecks:

- Limited *compute resources*
- Limited *on-chip memory capacity* for model parameters
- Limited *off-chip memory bandwidth*

→ Define a set of graph transformations to traverse the design space in a **fast** and **principled** manner

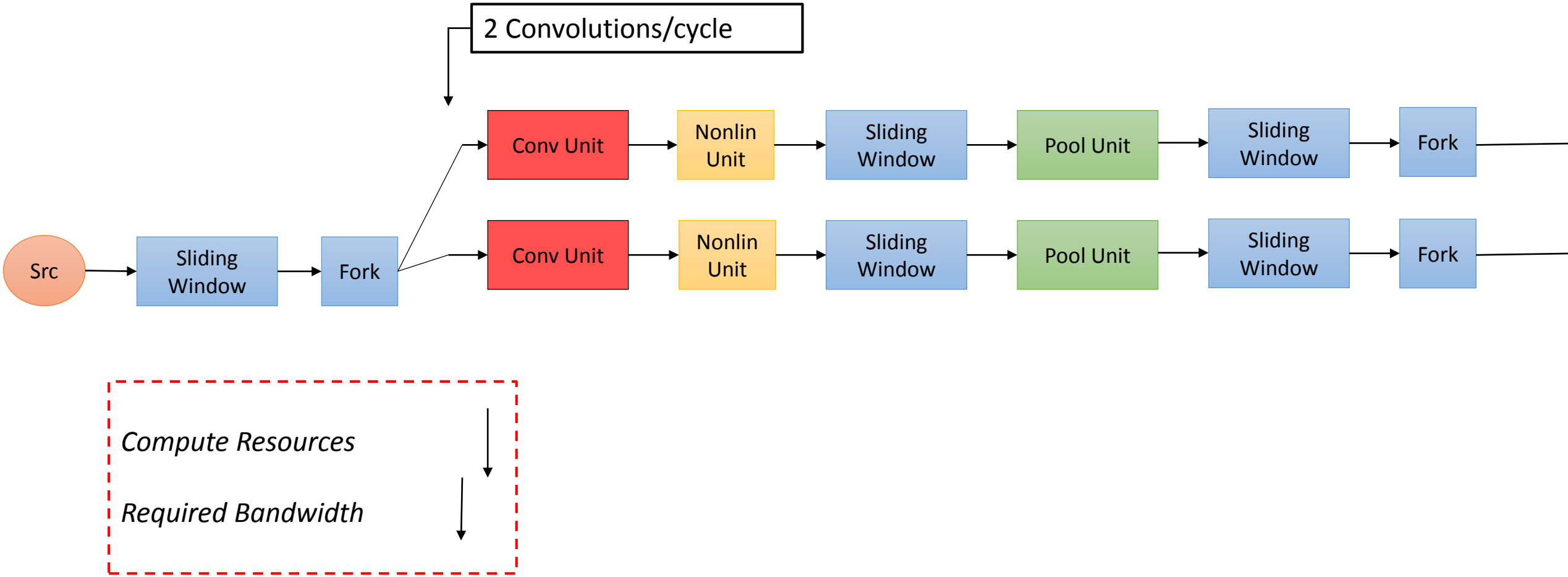
Transformation 1: Coarse-grained Folding



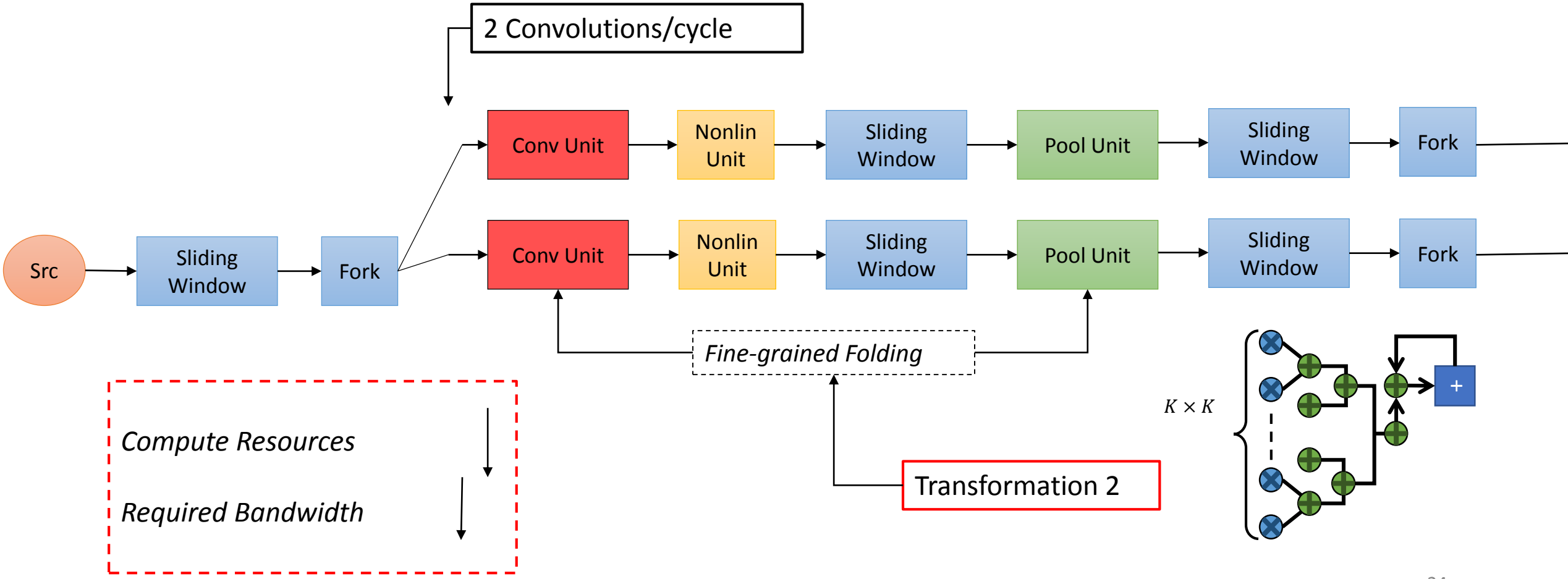
Exceeding the available compute resources

Not enough off-chip memory bandwidth

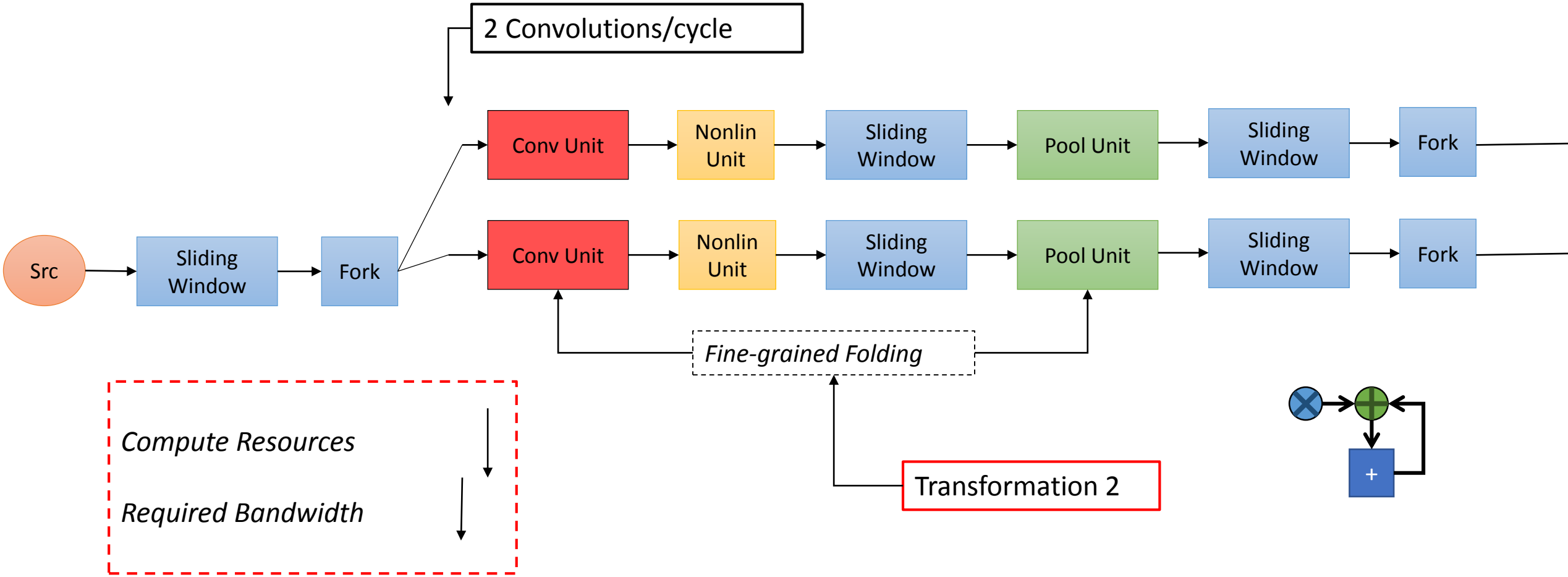
Transformation 1: Coarse-grained Folding



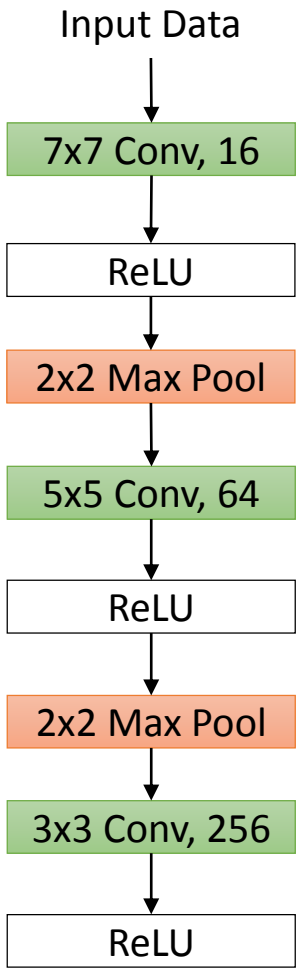
Transformation 2: Fine-grained Folding



Transformation 2: Fine-grained Folding



Transformation 3: Graph Partitioning with Reconfiguration

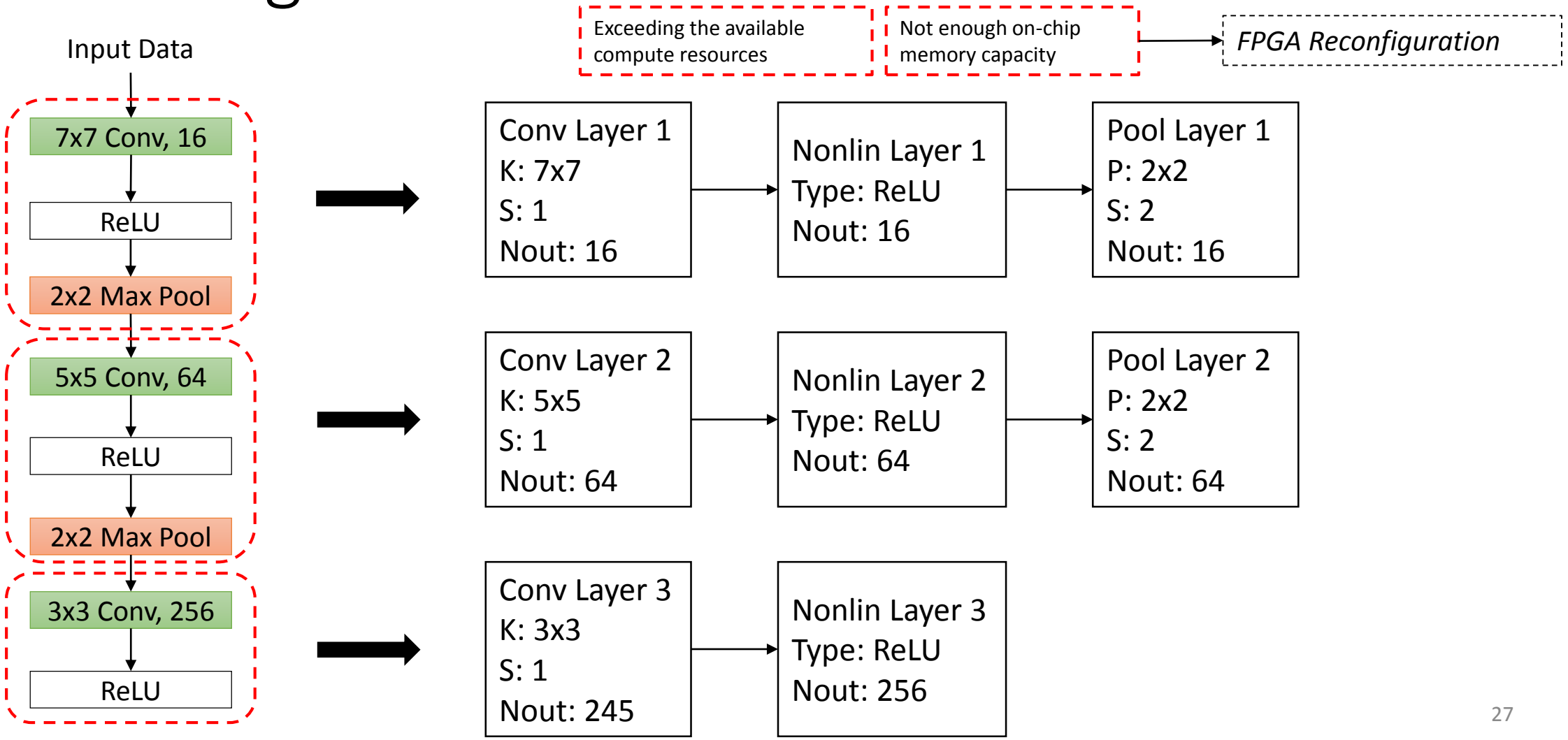


Exceeding the available compute resources

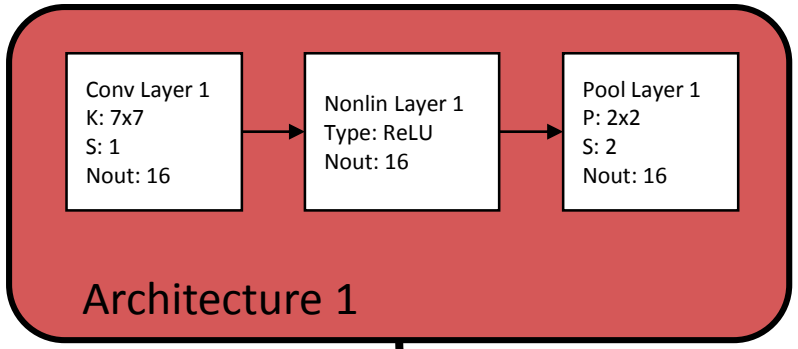
Not enough on-chip memory capacity

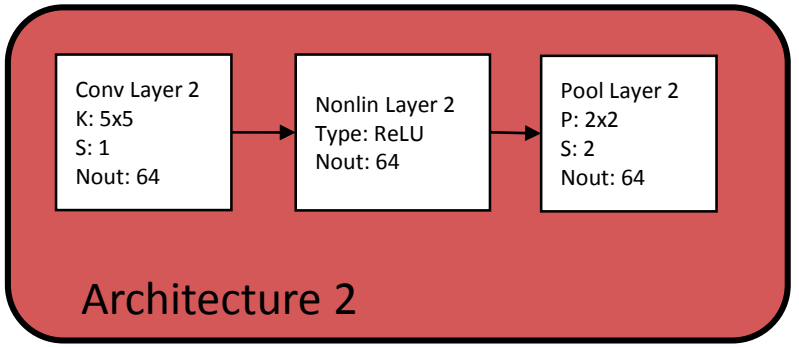
→ *FPGA Reconfiguration*

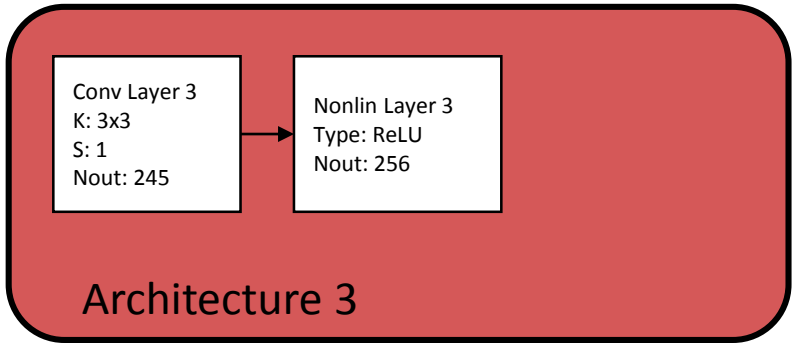
Transformation 3: Graph Partitioning with Reconfiguration



Transformation 3: Graph Partitioning with Reconfiguration

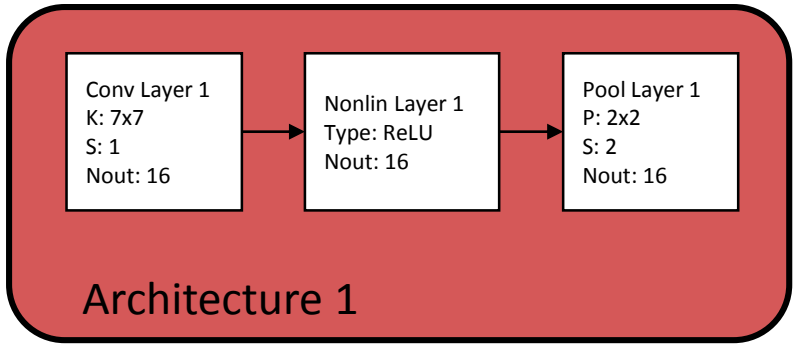


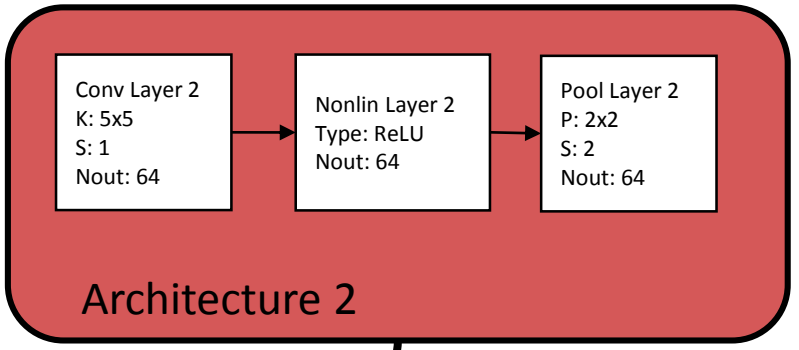


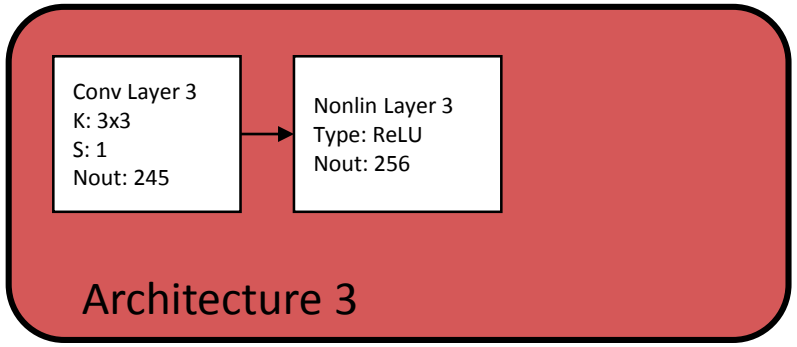


- Reconfigure FPGA
- Run network over batch
- Write-back to off-chip memory

Transformation 3: Graph Partitioning with Reconfiguration

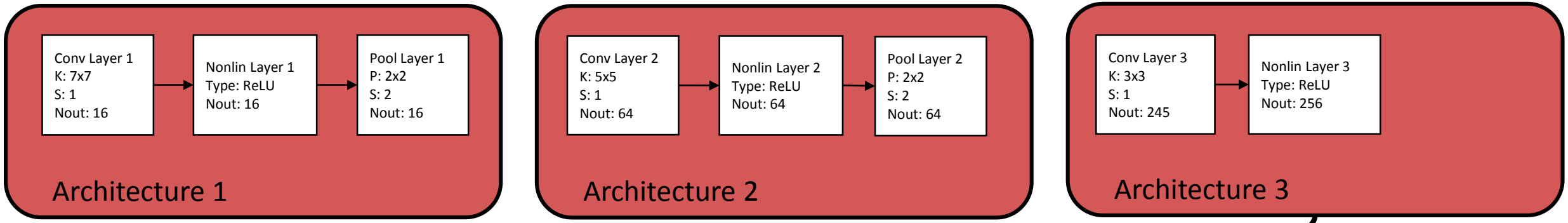






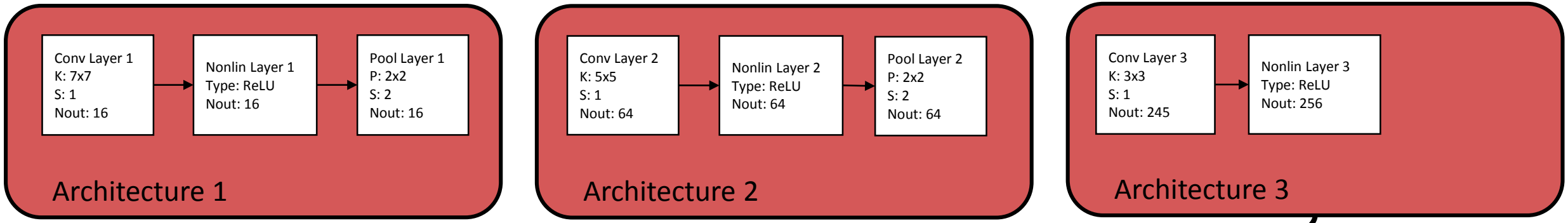
- Reconfigure FPGA
- Run network over batch
- Write-back to off-chip memory

Transformation 3: Graph Partitioning with Reconfiguration



- Reconfigure FPGA
- Run network over batch
- Write-back to off-chip memory

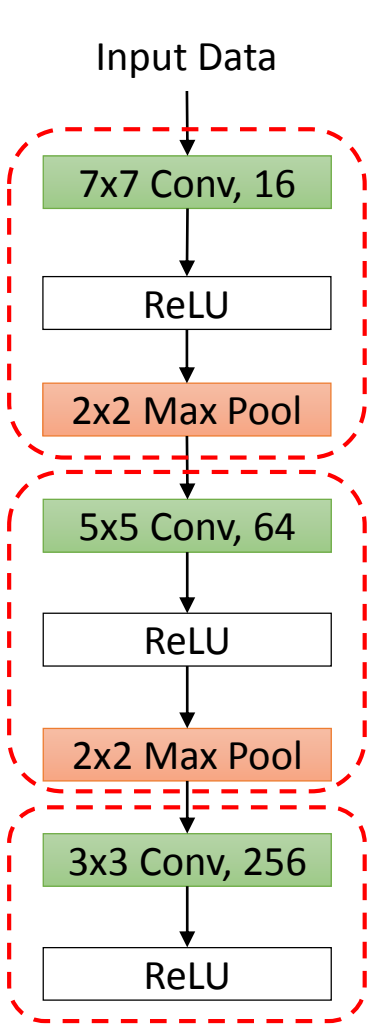
Transformation 3: Graph Partitioning with Reconfiguration



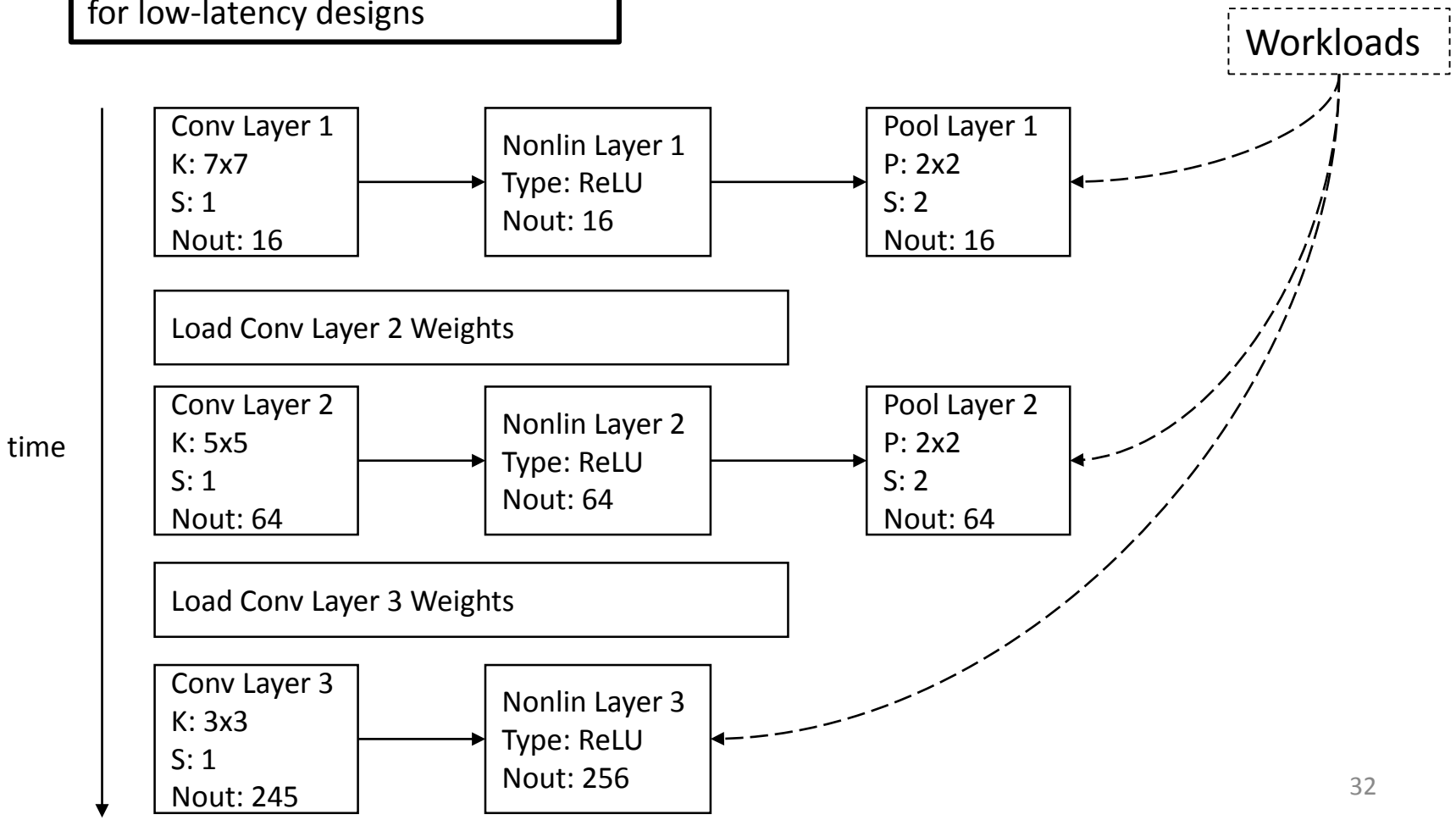
- Reconfigure FPGA
- Run network over batch
- Write-back to off-chip memory

- Batch processing amortises reconfiguration cost → high throughput
- Latency-sensitive applications?

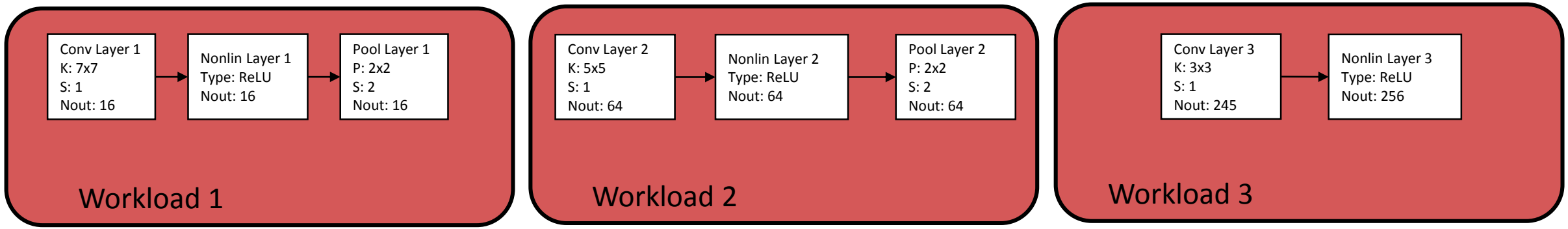
Transformation 4: Weights Reloading



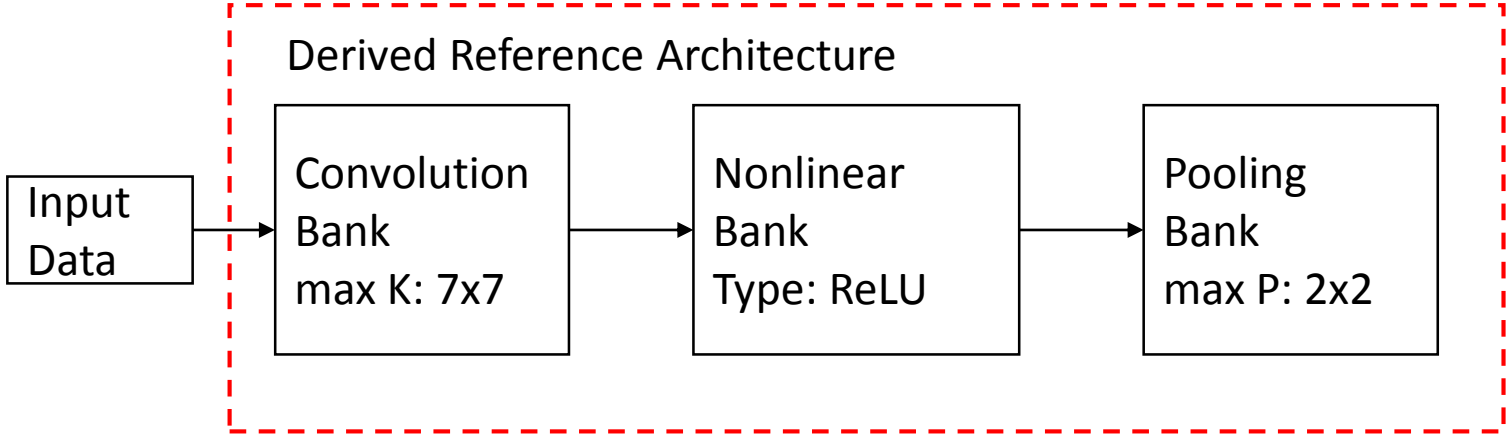
Run-time datapath reconfiguration for low-latency designs



Transformation 4: Weights Reloading

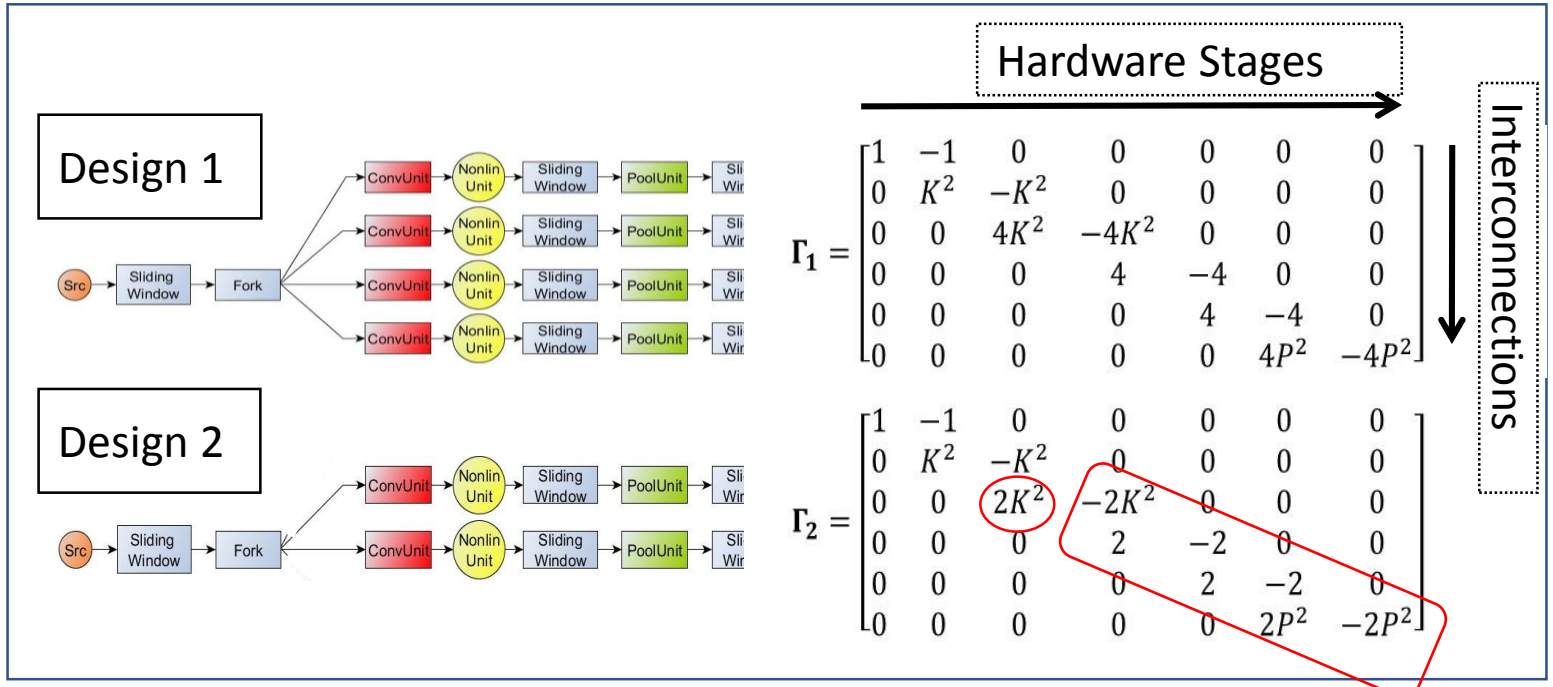


- Reload weights from off-chip memory and reconfigure datapath
- Run network over batch
- Write-back to off-chip memory

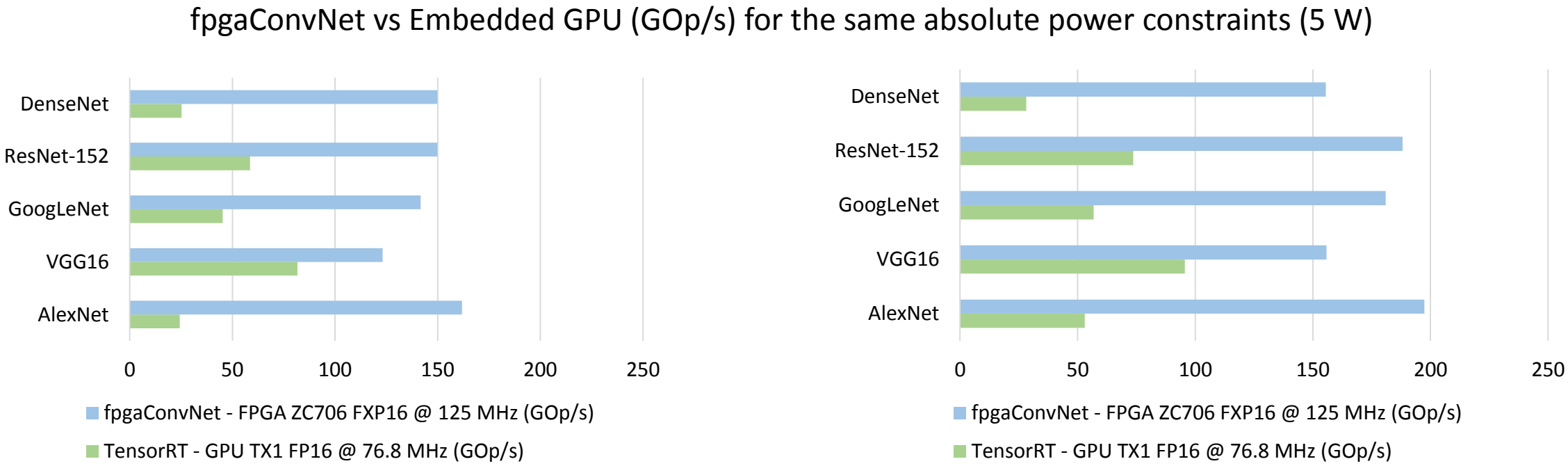


fpgaConvNet – Design Space Exploration and Optimisation

- SDF-based Framework
  - Capture hardware mappings as matrices
  - Transformations as *algebraic operations*
  - Analytical *performance model*
  - Cast design space exploration as a multiobjective optimisation problem
    - Maximise throughput
    - Minimise latency
    - Multiobjective criteria



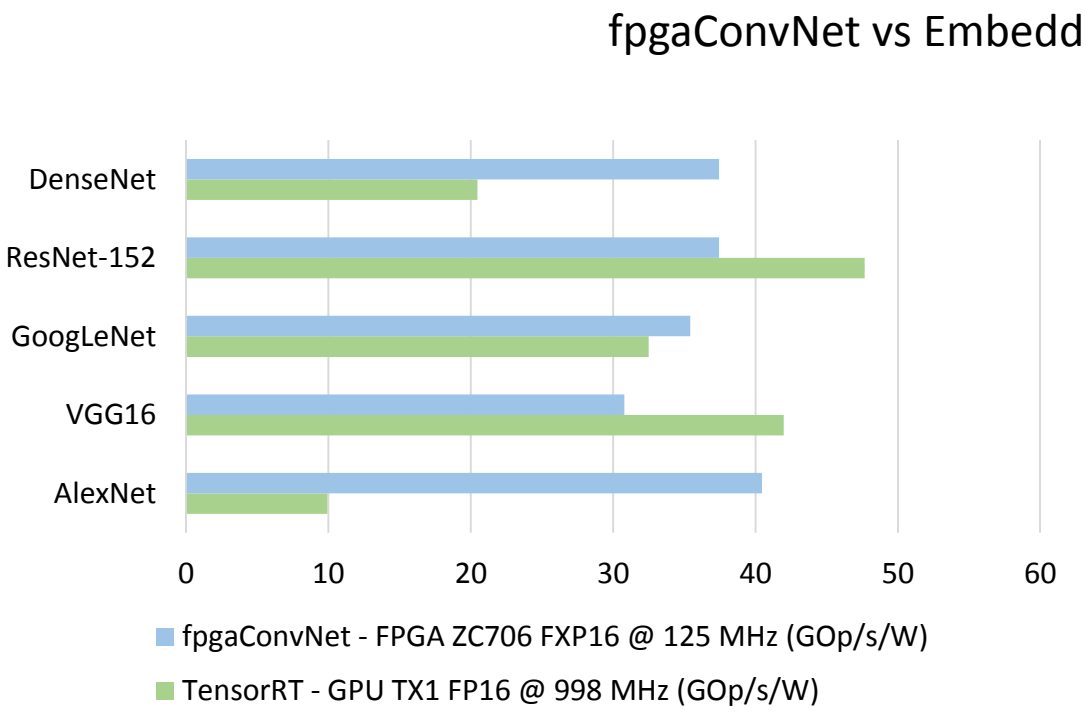
Comparison with Embedded GPUs: Same absolute power constraints

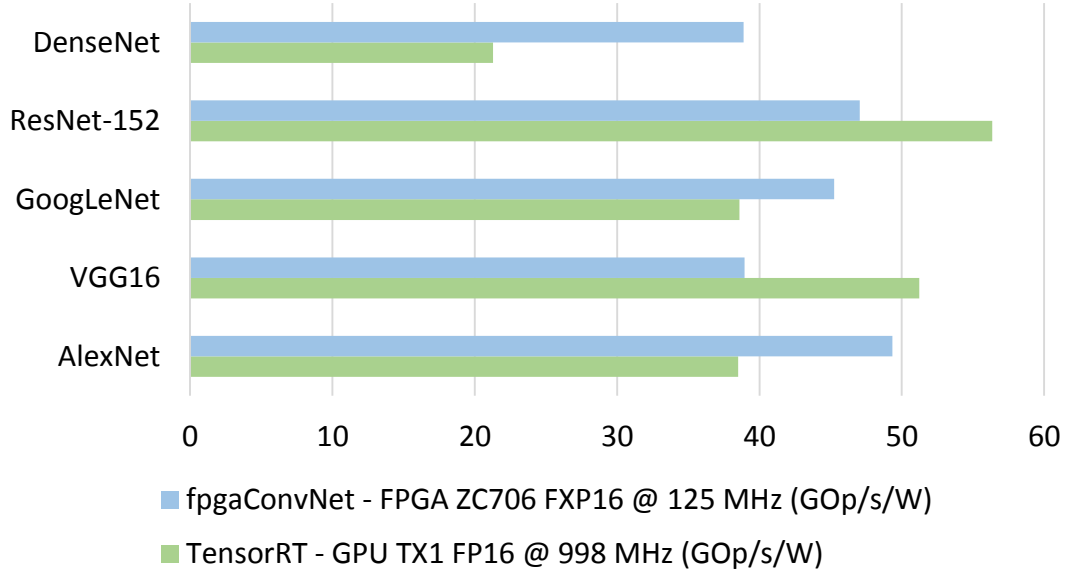


- Latency-driven scenario → batch size of 1
- Up to 6.65× speedup with an average of 3.95× (3.43 × geo. mean)

- Throughput-driven scenario → favourable batch size
- Up to 5.53× speedup with an average of 3.32× (3.07 × geo. mean)

Comparison with Embedded GPUs: Performance efficiency

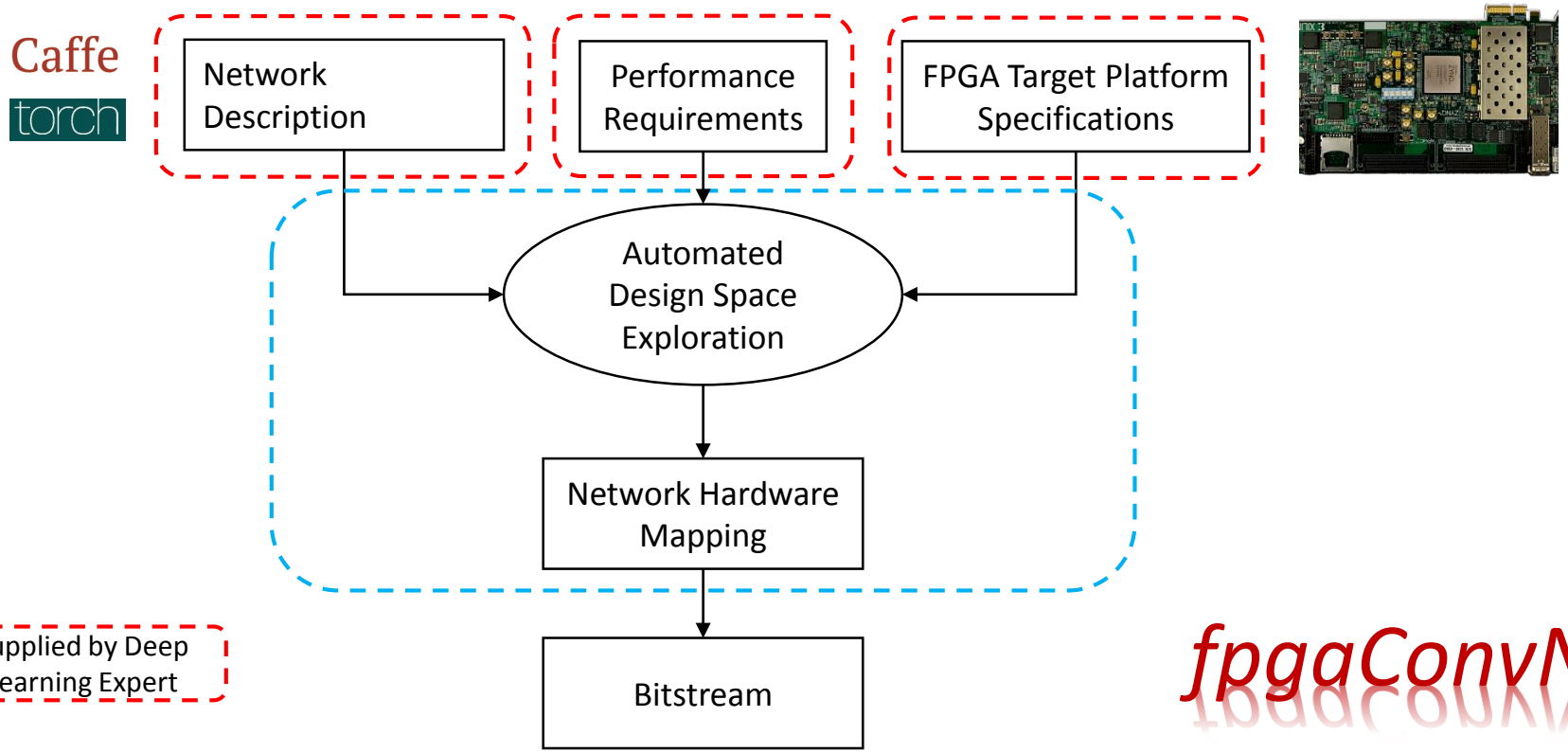




- Latency-driven scenario → batch size of 1
- Average of 1.70× (1.36× geo. mean) in GOp/s/W

- Throughput-driven scenario → favourable batch size
- Average of 1.17× (1.12× geo. mean) in GOp/s/W

Conclusion



More info: http://cas.ee.ic.ac.uk/people/sv1310/fpgaConvNet.html

Publications:
"fpgaConvNet: A Framework for Mapping Convolutional Neural Networks on FPGAs," *FCCM,* IEEE, 2016.
"Latency-Driven Design for FPGA-based Convolutional Neural Networks," *FPL*, IEEE, 2017.
"fpgaConvNet: A Toolflow for Mapping Diverse Convolutional Neural Networks on Embedded FPGAs," *NIPS 2017 Workshop on Machine Learning on the Phone and other Consumer Devices*, 2017.