Bridging The Gap Between Neural Network Exploration And Hardware Implementation

CEA- List {Nermine Ali, Michal Szczepanski, Suresh Pajaniradja, Pascal Vivet}
Siemens EDA {Ronan Poirier }

Context & Motivation

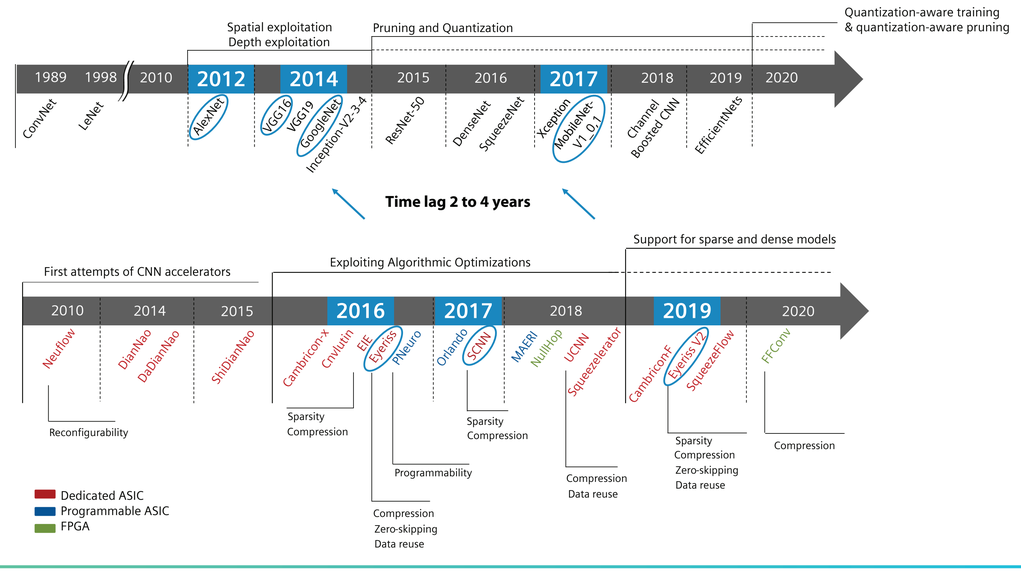
Deep learning (DL) is a fast growing research topic

- Continuous evolution of DL algorithms, i.e. new layers, hyper-parameters and topology of models
- Continuous optimization strategies, e.g. post and Aware training quantization, pruning, etc.

Hardware design cycle is quite significant

- Hardware designers struggle to master all algorithmic optimizations
  - Increasing complexity of design cycle
  - Multiple constraints: real-time constraints, memory and power constraints, etc.

How to bridge the gap between neural network exploration and hardware implementation ?



Objectives & Approach

- Fast and early DNN exploration of various deep neural network (DNN) models, applications and engines
  → Exploiting DL framework, **N2D2** [3]
- System sizing and exploration & IP block generation and RTL to GDS implementation
  → Exploiting High-Level Synthesis (HLS) tools, **Catapult HLS**
- Linking N2D2 tool to Catapult High-Level Synthesis tool
- Optimizing N2D2 C++ export to fit Catapult HLS engine requirements

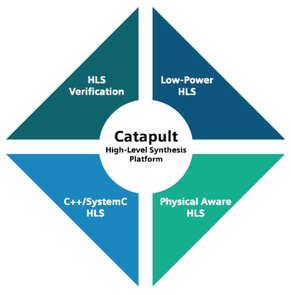
Early and fast exploration of complex & different AI models and generation of embedded IP for AI acceleration

- *Speed up the design process*
  - *An HLS-based flow is 4 times more productive compared to a standard RTL flow [1]*
- *Allow design space exploration of area vs performances tradeoff*

N2D2

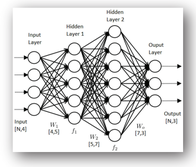
- Training and inference of DNN models
- Developing and optimizing DNN architectures
- Integrating innovative quantization/ compression/pruning algorithms
- Various export modules (C++, HW dedicated engines, HW for FPGA prototyping, Spiking NN, etc.)

Designing and verification at a higher level using Catapult HLS, which



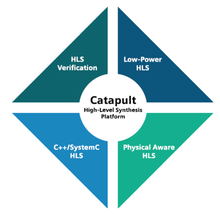
- Generates high quality ASIC and FPGA RTL from higher level C++ or SystemC
- Accelerates architecture exploration
- Proposes verification-centric from early C++/ SystemC development
- Runs RTL power analysis to get power numbers along with area and performance metrics

Tool Flow



1. DNN description
2. C++ description
3. RTL
4. GDS

N2D2



1. Selecting, training & Post-training Quantization to 8 bits
   Time: ~2h if CNN is Pre-trained
2. Exporting C++ description adapted to catapult (including testbench) & simulating for behavior verification
   Time: ~20min code export
3. HLS Synthesis with Catapult
   - Fast implementation of HW accelerators & streaming architectures for SoTA DNNs
   - Tradeoff throughput/performance vs. parallelism/area
   - Catapult generates RTL from C++
   - C++ testbench is used for simulation to verify the correct behavior of the design

   Time: few hours, varies with the DNN

MobileNet-V1 Model Generation & Exploration

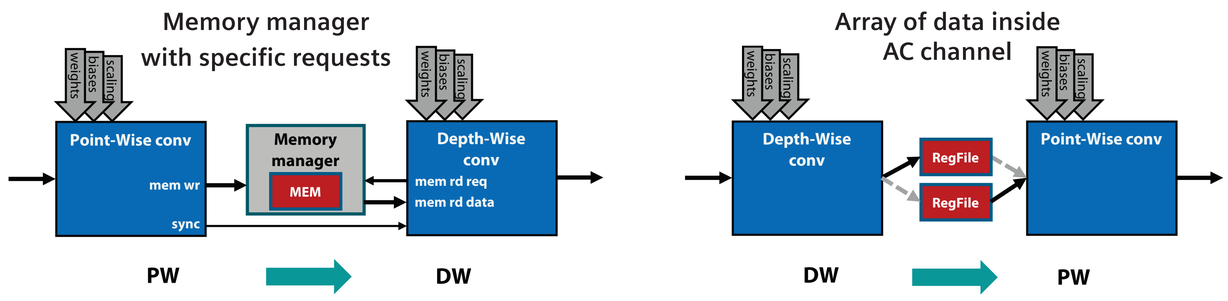
MobileNet-V1 [2] is a tunable Convolutional model

- width multiplier α=1
- Small memory footprints (4,2M parameters) & low computational requirements (0,56 GMACCs)
- Based on a sequence of Point-Wise (PW) and Depth-Wise (DW) layers

Training, 8bits Post-training Quantization & C++ export

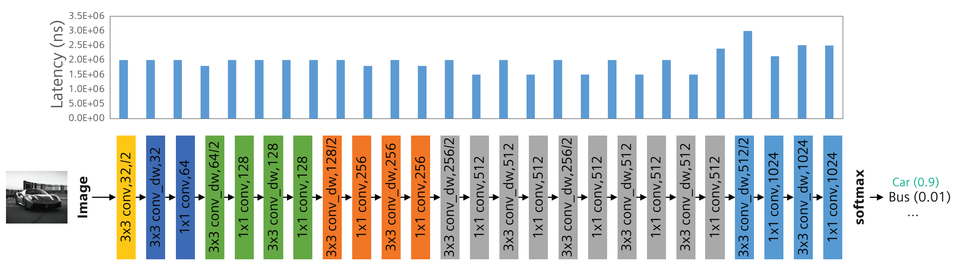
N2D2

Data transfers from **Point-Wise (PW)** **Depth-Wise (DW)** layers are implemented using optimized synchronizations and memory read and write queries



Results

- Regardless of the number of MAC operations, all layers are balanced in terms of execution time
- The proposed streaming architecture is pipelined and achieves a new frame every 400 000 cycles @5ns cycle time (500 fps) for a total execution latency (27 layers * 2 ms = 54ms)
- RTL simulation in Questasim PASSED, with C++ CNN golden model



Layers	Area (µm²)
1x1 conv, 512	252131
3x3 conv_dw, 512/2	20753
1x1 conv, 512	25753
3x3 conv_dw, 512/2	29058
1x1 conv, 1024	399413
3x3 conv_dw, 1024	22910
1x1 conv, 1024	628252

Conclusion & Acknowledgement

Proposal of new N2D2 Catapult-Stream export

- Where all layers are balanced in terms of execution time (cycles)
- Supporting complex CNN layers and Point-Wise ↔ Depth-wise data transfers
- Implementation with a data-flow block IP using ping-pong memory buffering scheme, with automatic synchronization

N2D2 stream-Catapult export has potential to become a leading HLS generator for large CNN networks

- Full automatic generation is ongoing, for full compatibility of C++ export with Catapult

NANOELEC. This work is partly funded thanks to the French national program "Programme d'Investissements d'Avenir", IRT Nanoelec" ANR-10-AIRT-05

[1] Sakari Lahti et al., « Are We There Yet? A Study on the State of High-Level Synthesis in: IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 38.5 (2019), pp. 898–911, doi: 10.1109/TCAD.2018.2834439
[2] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. CoRR abs/1704.04861 (2017). arXiv:1704.04861 http://arxiv.org/abs/1704.04861
[3] N2D2 is available Open-Source at https://github.com/CEA-LIST/N2D2

60 DESIGN AUTOMATION CONFERENCE