Energy-efficient Object Detection for Autonomous Vehicles

Marcel Flottmann, Michael Mecik, Martin Kumm

Motivation

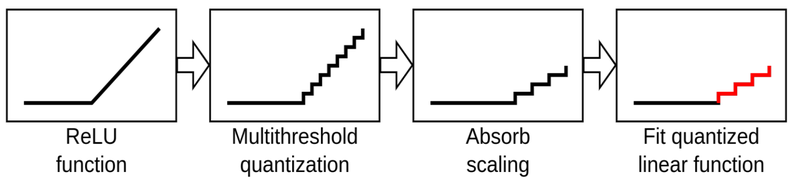
Artificial Intelligence (AI) provides key technologies for the implementation of autonomous vehicles, especially when it comes to identifying objects in the environment, such as other vehicles, pedestrians or road signs.

- **Requirements:** Real-time processing and low power consumption
- **Problem:**
  - Image classification and object detection networks are computationally intensive
  - Field Programmable Gate Array (FPGA) accelerators show promising results in implementing DNN inference, but no state-of-the-art DNN for object detection is currently available
- **Objectives:**
  - Low power object detector for the Campus FreeCity project
  - Integration into the EDAG CityBot

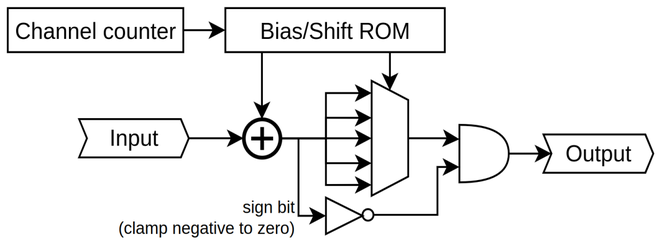
Contribution

The following contributions have been implemented:

- **Implementation of YOLOv6**
  - mAP50: 68,5% on VOC dataset and 6-bit quantization
  - Custom Transformations added to transform SPPF layer, fork/join operations
- **PyVerilator improvements for faster simulations**
  - Reduced computation overhead
  - Use multithreading
- **Packaging IPs for large block designs**
  - Vivado re-parses all previously added Verilog sources
  - Package nodes as individual IP blocks
  - From quadratic to linear time
- **Power-of-two Quantization with Brevitas**
  - Custom quantization with restriction of scaling factors to powers of two
  - Quantization-aware training can compensate loss of precision
- **Power-of-two linear activation**
  - ReLU is used as activation function and converted to MultiThreshold in FINN
  - Scaling factors are absorbed and change slope of the ReLU function
  - Power-of-two factors can be implemented as a simple shift
  - Each layer needs only a few values to shift by, therefore multiplexers can be used



Transformation of Multithreshold to power-of-two linear activation



Block diagram of the power-of-two linear activation implementation

Key Ideas

We extended FINN to support large artificial neural networks:

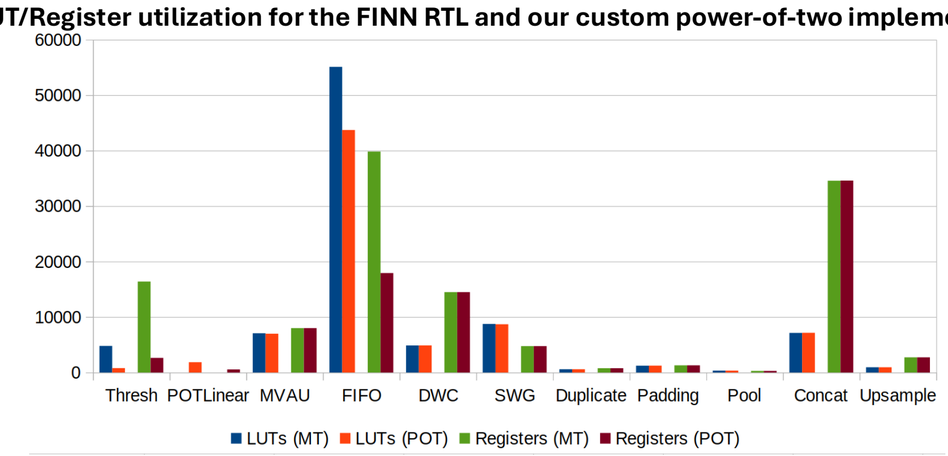
- **Optimized toolflow**
  - Identified bottlenecks and limitations
  - Reduce build time
- **Optimize FINN nodes**
  - MultiThreshold vs power-of-two linear (POT) activation
  - Reduce resource usage by using FPGA-friendly operations
- **Add new transformations for streamlining**
  - Generic transformations are not sufficient for e.g. SPPF and other specific fork/join topologies

Results

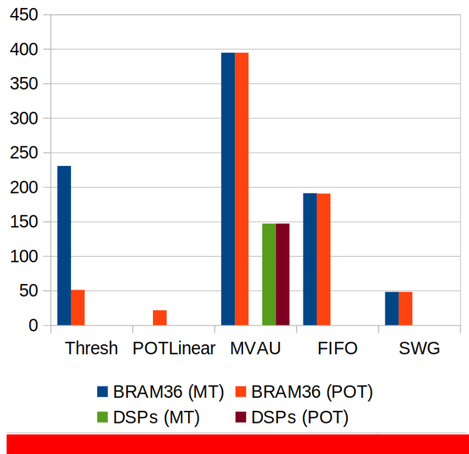
Iterations of improvement with FPS=30, Clock=200MHz , UltraScale+ Architecture

	Alveo U280		ZCU102		
Resource	**Initial**	**FIFO Bug**	**Power-of-two HLS**	**FINN RTL**	**Power-of-two RTL**
LUTs	559,463	276,009	139,960	122,729	109,134
Logic	415,560	243,164	116,996	95,243	82,118
LUTRAM	56,538	6,757	1,641	3,347	3,347
SRL	87,365	26,088	21,323	24,139	23,669
Flip-Flops	667,996	416,743	245,234	186,983	152,022
BRAM18	1,082	1,770	1,024	1,744	1,427
URAM	5	15	0	0	0
DSPs	1,198	189	181	149	149
Description	Unmodified FINN repository	FIFO sizing led to an overflow reducing performance	Implemented power-of-two optimization in HLS	Newer FINN version introduced MVAU and Thresholding in RTL	Implemented power-of-two optimization in RTL

LUT/Register utilization for the FINN RTL and our custom power-of-two implementation



DSP/BRAM utilization for the FINN RTL and our custom power-of-two implementation



Demonstrator Comparison RTX2080 vs. ZCU102

Future Work

- **Short-term goals**
  - Improving YOLOv6 mean average precision (mAP)
  - Reduce size of data management components like FIFOs, DWC, etc.
- **Long-term goals**
  - Adding further custom transformations and nodes to FINN to implement newer YOLO versions

Discussion

- Complex neural network need fine-tuned and/or custom transformations
- Most of the FPGA resources are used for data management (FIFO, data width converter, etc.)
- Power-of-two may be a too strict constraint that impacts the neural networks performance metrics: ordinary linear activation with a DSP would also save resources

Contact: <firstname>.<lastname>@informatik.hs-fulda.de