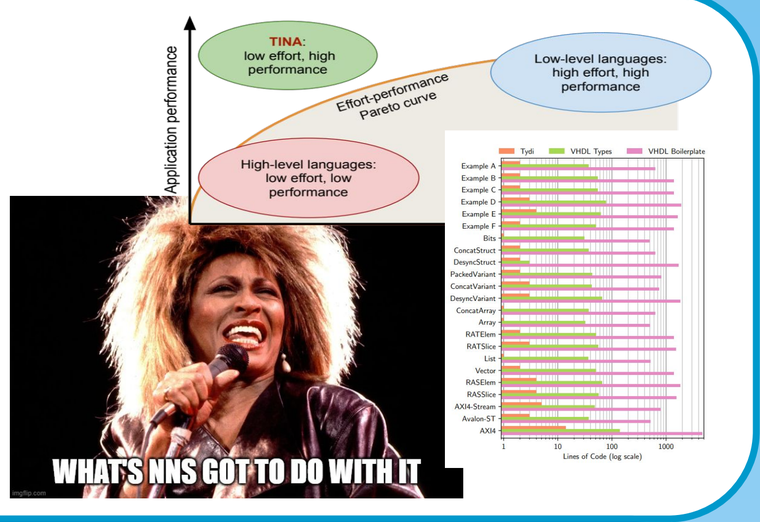
Running non-NN algorithms on an FPGA using FINN and TINA

C. Boerkamp[@1] Z. Al Ars[1] S. van der Vlugt [2]

[@]C.Boerkamp@tudelft.nl [1]Delft Quantum and Computer Engineering, TU Delft [2]ASTRON

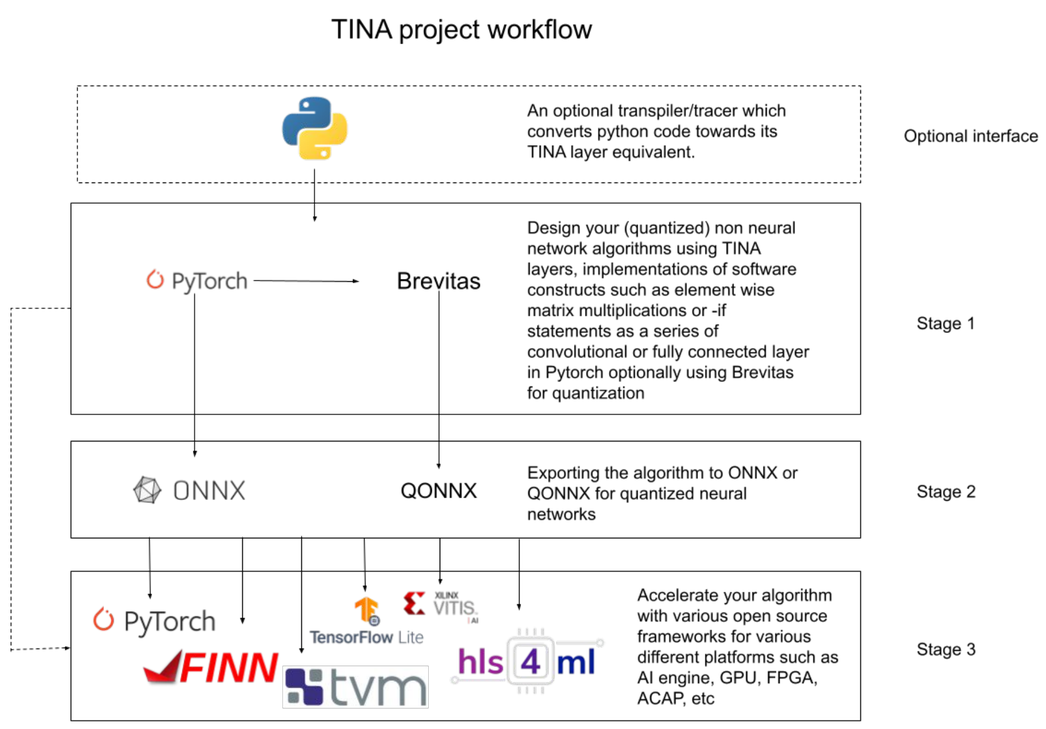
1 Introduction and motivation

- Programming FPGAs is often low-level, demanding deep hardware knowledge.[1,2,3]
- The code for CPU/GPU and other hardware does not work for FPGAs and vice versa leading to a fragmented software ecosystem.
- Limits efficiency and restricts the application domains.
- This is where TINA comes in: A high level language based on Python.[5]



2 What is TINA

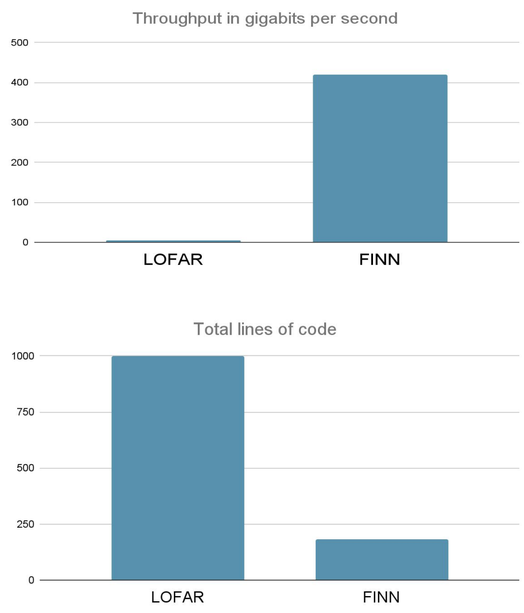
- TINA is based on the idea that almost every operation can be mapped towards a series of convolutions, the so called TINA layers.
- Using the ubiquity of convolutions we can accelerate our TINA layers on any device that supports NNs.
- Using ONNX as our conversion medium allows us to leverage most frameworks for accelerating NNs for various types of hardware.
- Right now algorithms are made by a series of TINA layers, however in the future we wish to create a transpiler that automatically converts code into TINA layers.
- We have already accelerated algorithms on a GPU[5] and on an NPU[6] now we will try to accelerate on an xc7z020clg400-1 FPGA using FINN.



3 Preliminary results

- As a point of reference, we compared our FINN implementation of a Polyphase Filter Bank running on an xc7z020clg400-1 FPGA to an implementation for the LOFAR radio telescope [4] running on an Intel Arria 10 FPGA.
- We can easily convert the TINA layers towards Brevitas which, combined with automatic buildflow, allows us to create computationally complex FPGA accelerated algorithms in 184 lines which is around 80% reduction in the amount of code as compared to the LOFAR implementation, which is written in VHDL [7].
- The throughput with our implementation through TINA achieves 209 Gb/sec which is a few factors higher than the throughput of the LOFAR implementation of 3.88 Gb/sec (this last throughput is requirement driven, not the top performance of the implementation).
- In return the resource usage from our implementation is a factor higher.

	FINN	LOFAR [4]
BRAM	1020	56
FFs	39268	47701
LUTS	37507	20685
DSP	264	54
URAM	0	0



4 Conclusions

- We integrated TINA within both Brevitas as well as FINN needing very few adjustments towards existing code.
- We demonstrated that with TINA we can easily target different devices such as the GPU[5] and on an NPU[6] and also an FPGA
- Generated accelerators using FINN and TINA are highly competitive as compared to low-level language implementations on FPGA, resulting in this comparison in **5x less code** and **50x more throughput**

5 Future work

- Create an automated flow of FINN towards Brevitas using its post quantisation training functions.
- Create various accelerated implementations of algorithms using TINA in combination in FINN in order to find where the limitations in both frameworks lay.
- More extensive comparison to reference implementations.

Acknowledgement

This project has received funding from the Eureka Xecs TASTI project (grant no. 2022005), Horizon Europe research and innovation program RADIOBLOCKS project (grant no. 101093934) and the Netherlands eScience Center RECRUIT project.

References

1. J. Hoozemans, J. Peltenburg, F. Nonnemacher, A. Hadnagy, Z. Al-Ars and H. P. Hofstee, "FPGA Acceleration for Big Data Analytics: Challenges and Opportunities," in IEEE Circuits and Systems Magazine, vol. 21, no. 2, pp. 30-47, Secondquarter 2021, doi: 10.1109/MCAS.2021.3071608.keywords: {Field programmable gate arrays;Computer architecture;Standardization;Reconfigurable logic;Big Data;Throughput;Hardware},
2. Cromjongh, Casper, et al. "Tydi-Chisel: Collaborative and Interface-Driven Data-Streaming Accelerators." 2023 IEEE Nordic Circuits and Systems Conference (NorCAS). IEEE, 2023.
3. Yu, Hejie, et al. "Implementation of Convolutional Neural Network with Co-design of High-Level Synthesis and Verilog HDL." 2020 IEEE 15th International Conference on Solid-State & Integrated Circuit Technology (ICSICT). IEEE, 2020.
4. Schoonderbeek, G. W., et al. "UniBoard2, A generic scalable high-performance computing platform for radio astronomy." Journal of Astronomical Instrumentation 8.02 (2019): 1950003.
5. Christiaan Boerkamp, Steven van der Vlugt and Zaid Al-Ars, "TINA: Acceleration of Non-NN Signal Processing Algorithms Using NN Accelerators", IEEE Int'l Workshop on Machine Learning for Signal Processing (MLSP), 2024
6. https://www.hackster.io/tina/tina-running-non-nn-algorithmns-on-an-amd-ryzen-npu-0cc58c
7. https://git.astron.nl/rtsd/hdl/-/tree/master/libraries/dsp/wpfb

TU Delft Delft University of Technology

TASTI

ASTRON