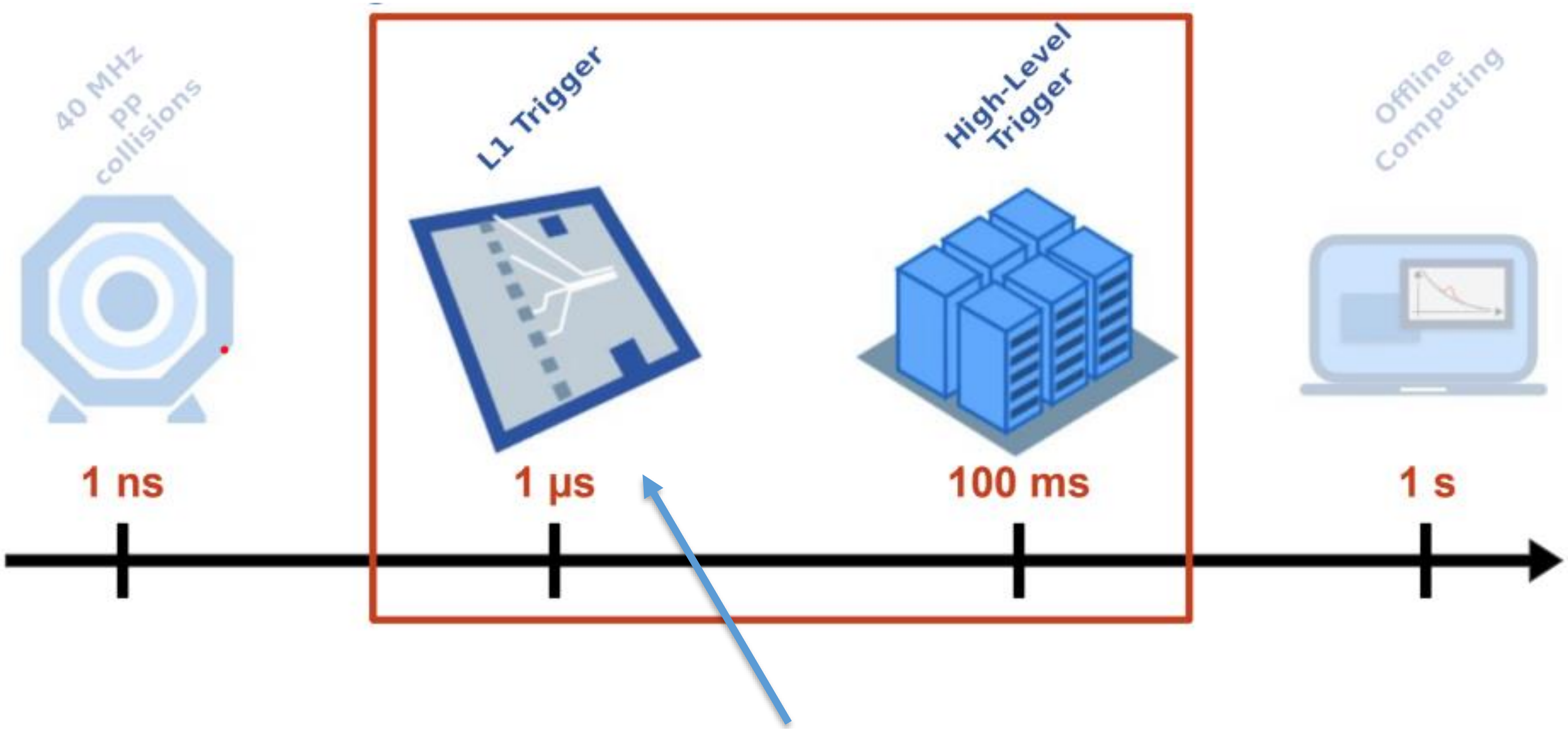
Faster FPGA firmware synthesis with hls4ml

FastML group

Sarai Sokolovsky
Supervised by Vladimir Loncar

THE LHC BIG DATA PROBLEM

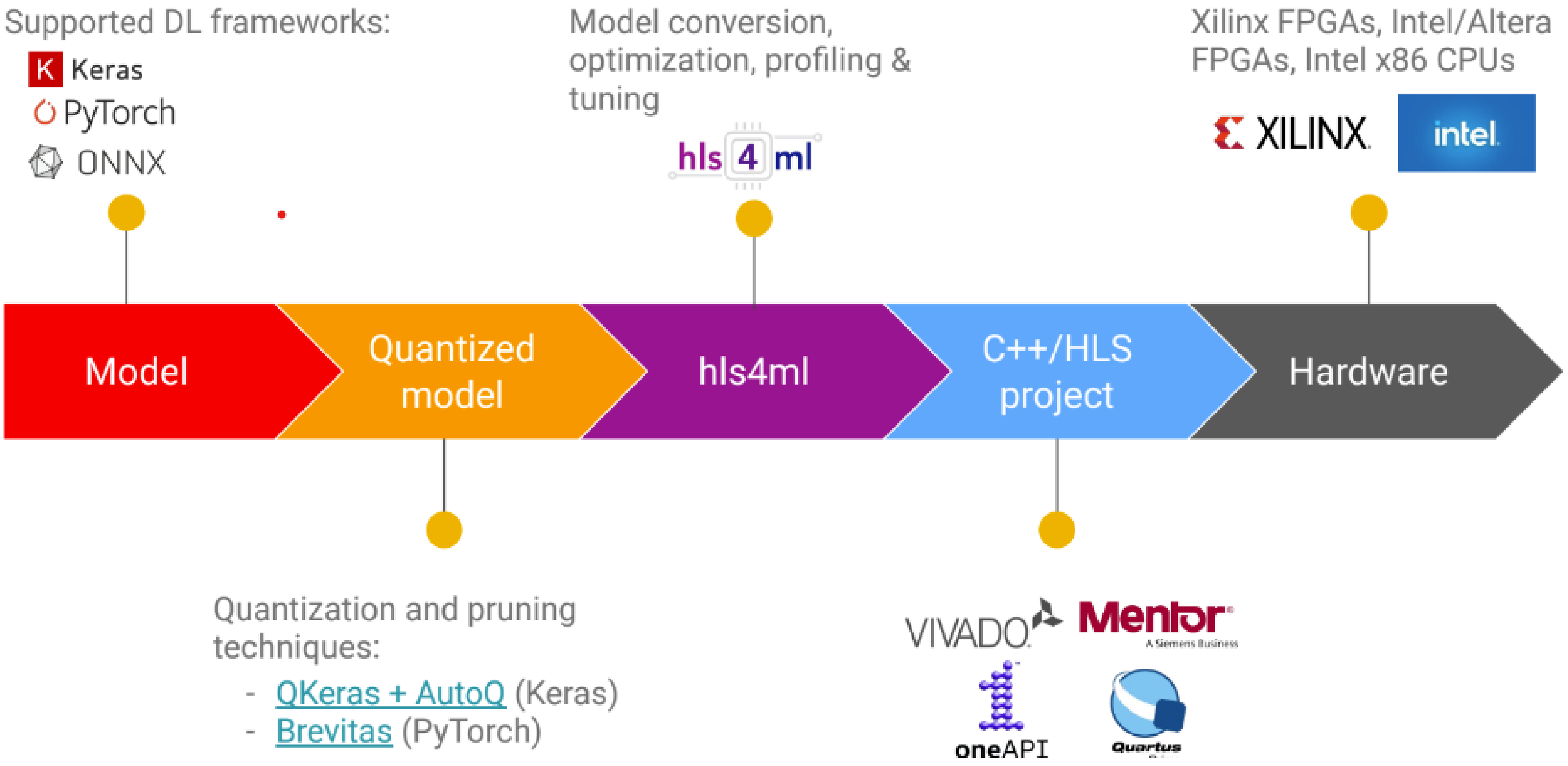
Deploy ML algorithms very early
Challenge: strict latency constraints!



- *Fast processing of raw data*
- *Flexibility and modularity*

Field-programmable gate arrays (FPGAs

- ✓ *Reprogrammable integrated circuits*
- ✓ *Massively parallel = low latency*
- ✓ *Low power*



Supported DL frameworks:

- Keras
- PyTorch
- ONNX

Model conversion, optimization, profiling & tuning

Xilinx FPGAs, Intel/Altera FPGAs, Intel x86 CPUs

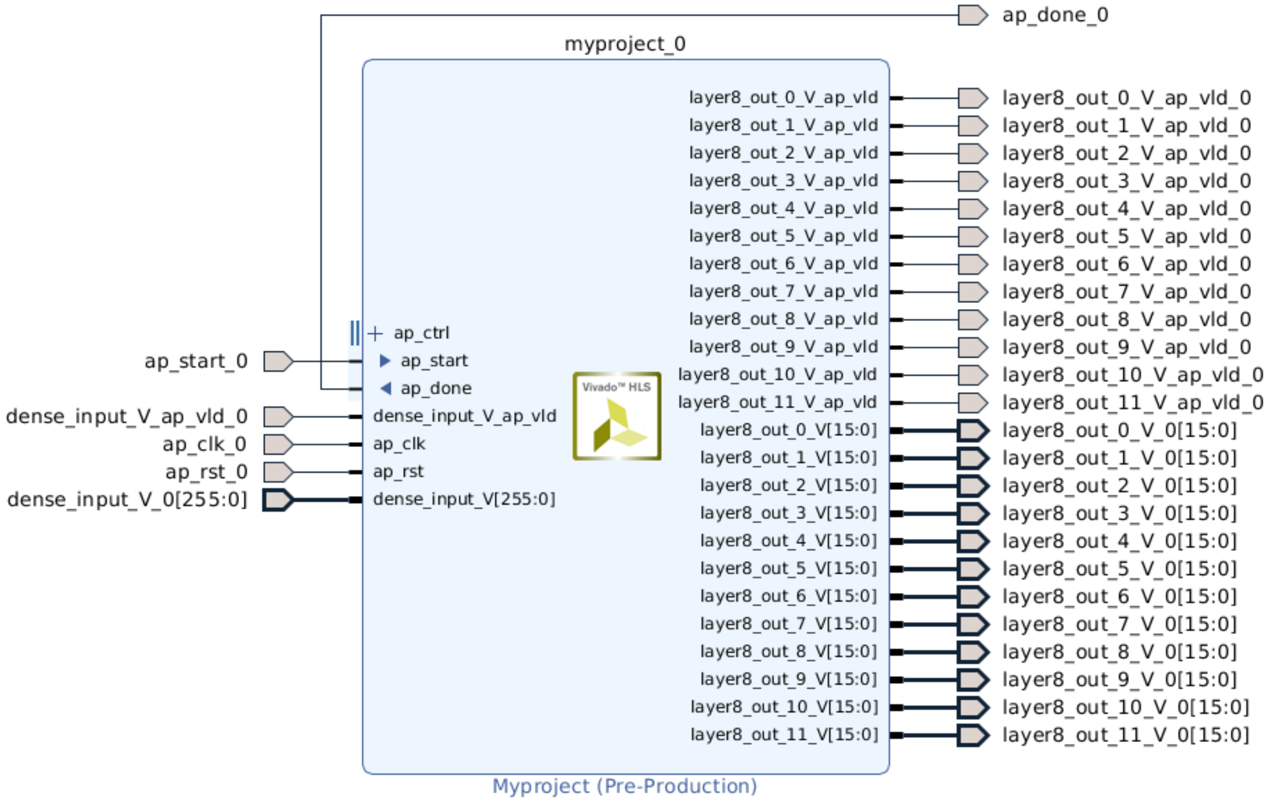
Model → Quantized model → hls4ml → C++/HLS project → Hardware

Quantization and pruning techniques:

- QKeras + AutoQ (Keras)
- Brevitas (PyTorch)

CURRENT ARCITECTURE

inputs



outputs

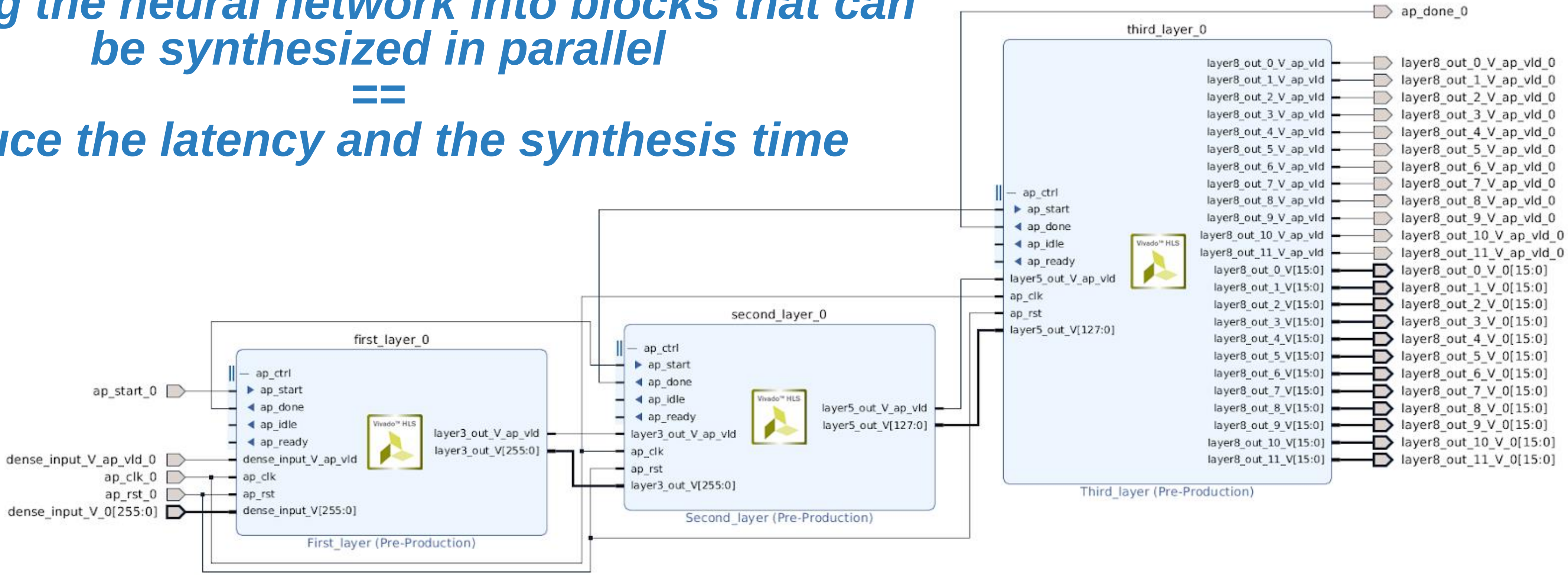
module

SOLUTION

dividing the neural network into blocks that can be synthesized in parallel

==

reduce the latency and the synthesis time



Less gates -> less output capacitance -> less time for the signal to go through

IMPROVEMENTS

Reduced by ***2x-20x!***

Reduced by ***7%!***

	Synthesis (s)	*Latency (ns)*	*Power (W)*	*Total BRAM*	*Total DSP*	*Total LUT*	*Total FF*
old	***50***	*145*	*464*	*3*	*408*	*16149*	*11656*
new	***20-30***	*135*	*459*	*3*	*408*	*16149*	*11656*

Same amount!

THANK YOU!