Expanding the hls4ml Backend for PandA/Bambu

Shahd Abuhassan | CERN openlab 2026 | Edge SpAIce

How a network becomes hardware

compile(): the software check

g++ builds the generated C++ as an ordinary program. predict() runs it on the CPU and the numbers are checked against Keras. Fast, and no HLS tool is involved.

build(): the hardware path

Bambu synthesizes the same C++ to Verilog; the RTL is simulated and placed on the chip. Slow. This is the path the project exists for, and the one I verified.

Why a new backend?

hls4ml's existing flow is vendor-specific

Today the path runs through AMD's Vitis HLS. Bambu (Politecnico di Milano) is an open, vendor-neutral HLS compiler: one model can target any characterized FPGA. This opens the door to future ML applications at the LHC and the FCC.

A satellite that finds plastic in the ocean using hls4ml

Edge SpAIce detects marine litter from orbit. Downlink is scarce, so the network has to run on board.

A European space-grade FPGA

The target is NanoXplore's NG-ULTRA: radiation-tolerant, but with no native NanoXplore HLS tool.

My role: verify the Bambu backend end-to-end across the hls4ml test suite.

What I did

The verification

- 9 test files, 142 test cases under --backend-filter=Bambu
- Official CI image, pinned by digest
- Run on corellia, the same machine the CI pipeline itself uses
- Bambu ginevra4 (dev/panda), reproducible setup scripted

Four gaps in the CI image

- Missing libraries: xelab, Vivado's simulator, cannot load
- Missing en_US.UTF-8 locale: xelab aborts
- Vivado crashes in containers after routing (a documented AMD issue), turning passes into false negatives
- Image pins an old Bambu that reproduces a fixed bug

All four diagnosed, worked around, and documented. **The fix list is going into the shared CI image.**

Results

Test file	Result
test_build_bambu (synth, cosim, vsynth)	7 / 7 ✓
test_activations	36 / 36 ✓
test_auto_precision	4 / 4 ✓
test_dense_unrolled	20 / 20 ✓
test_keras_api	10 / 10 ✓
test_multi_dense	16 / 16 ✓
test_pooling	3 / 12 ✗
test_softmax	12 ✓ + 24 documented skips
test_report (Bambu XML parser)	1 / 1 ✓

109 cases pass, 9 fail, 24 documented skips.

The one red file, two causes:

- pool2d: Bambu front-end failure. Confirmed by Bambu's author; a fix is in progress.
- pool1d, padding='valid': ~70,300 LUTs vs 13,638 for padding='same'. Five times the hardware, and it doesn't fit the chip. Reported upstream.

What's next

A green Bambu CI job on the CERN runner

The pipeline definition exists; it needs the environment fixes and a Bambu version bump.

The CI image gets its four fixes

Now being integrated into the shared image, working from the documented list this verification produced.

Investigate io_stream (streaming) for pooling

My next task: enabling the currently-excluded streaming tests one by one.

Handover

Environment guide, full results, and open findings, so the next person starts exactly where this work stops.

CERN openlab

One model, any FPGA.

A vendor-neutral hls4ml opens targets no single vendor covers: commercial, space-grade, or fully custom hardware. Edge SpAIce is the first passenger; it will not be the last.

Verified: 8 of 9 test files green. Two upstream bugs found, reported, and in the developers' hands.

Thank you | Questions?

Backup: the pooling failures in detail

pool1d, io_parallel: post-synthesis LUTs (xc7a100t has 63,400)

configuration	LUTs	outcome
max, padding=same	**13,638**	pass
max, padding=same, strides=4	**13,638**	pass
avg, padding=same	**18,283**	pass
max, padding=valid	**70,153**	fail: DRC UTLZ-1, placer refuses
avg, padding=valid	**70,456**	fail: DRC UTLZ-1
avg, padding=valid, strides=1	**70,333**	fail: DRC UTLZ-1

Same layer, output length 41 vs 42 elements, yet a 5x LUT difference between padding modes. Simulation and synthesis succeed; only placement fails.

pool2d, io_parallel: all 6 variants

- Exit code 11 at Frontend::CreateIRManager
- "Front-end compiler returns an error during compilation 38", no clang diagnostic printed
- Identical for max/avg, valid/same, and both strided variants
- Bambu's author reproduced it: clang runs out of memory. A front-end rework in progress at Politecnico should resolve it.

```
INFO: [Route 35-16] Router Completed Successfully
ERROR: [DRC UTLZ-1] Resource utilization: LUT as Logic over-utilized in Top Level Design.
This design requires 70153 of such cell types but only 63400 compatible sites are available.
```

Backup: the four CI-image fixes

libtinfo5 / libncurses5 missing

xelab fails: "error while loading shared libraries: libtinfo.so.5". Breaks cosim and vsynth. Fix: install the .debs from archive.ubuntu.com.

en_US.UTF-8 not generated

Vivado's rdiArgs.sh forces LC_ALL=en_US.UTF-8; xelab aborts with "locale::facet::_S_create_c_locale name not valid". Fix: locale-gen en_US.UTF-8 (C.UTF-8 is not enough).

Vivado SIGABRT in libudev after routing

Crash in the WebTalk/license host scan, after all reports are written, so a success reads as a failure. Documented AMD issue (article 000034450). Fix: LD_PRELOAD=/lib/x86_64-linux-gnu/libudev.so.1, set per process as AMD recommends.

setup-bambu.sh pins an old Bambu

The script downloads ginevra2 (c06bb77e), which reproduces the io_stream "Function not found" bug that ginevra4 fixes. It also caches the extraction under ~/.tools/bambu and silently ignores BAMBU_APPIMAGE unless that cache is removed.