Analiza performantei (FPGA Synthesis & Performance Evaluation)

1. Scop si metodologie

Obiectiv. Evaluarea implementarii pe FPGA a acceleratorului QNN prin rapoarte de sinteza/implementare (resurse, timing, power) si estimarea performantei (latenta, throughput), urmata de comparatia cu un baseline software (CPU/GPU pe Windows).

Instrumente folosite.

- **Vivado**: rapoarte post-implementation (utilizare resurse, timing, consum).
- **Verilator**: simulare cycle-accurate pentru masurarea numarului de cicluri/inferenta.
- **Windows + PyTorch**: baseline pe CPU si pe GPU (RTX 3050 Ti).

Modul top evaluat.

- Top FPGA: qnn_fpga_wrapper (I/O redus: clk, rst_n, start, done), care contine intern qnn_accel_top.

2. Rezultate FPGA (post-implementation)

2.1 Utilizarea resurselor

Tinta: **xc7a35tcpg236-1 (Artix-7)**
Din raportul Vivado (utilization):

- **Slice LUTs:** 61
- **Slice Registers (FF):** 78
- **BRAM (Block RAM Tile):** 0
- **DSP blocks:** 0
- (optional) **Bonded IOB:** 4, **BUFGCTRL:** 1

Interpretare. In configuratia curenta, designul foloseste foarte putine resurse si nu utilizeaza blocuri DSP sau BRAM dedicate.

2.2 Timing si frecventa maxima

Din “Timing Summary” (Vivado):

- Constrangere clock: **10 ns** (100 MHz)
- **WNS (setup): 4.941 ns**
- **TNS: 0.000 ns**, failing endpoints: 0
- Status: **All user specified timing constraints are met.**

Estimare Fmax.
Perioada minima: **Tmin ≈ 10 ns − 4.941 ns = 5.059 ns**
Rezulta: **Fmax ≈ 1 / 5.059 ns ≈ 198 MHz**

2.3 Consum de putere (estimare Vivado)

Din raportul de power (vectorless):

- **Total On-Chip Power:** 0.071 W
- **Static:** 0.070 W (~98%)
- **Dynamic:** 0.001 W (~2%)
- Confidence: **Medium** (switching activity vectorless)

Observatie. Fiind estimare vectorless (fara SAIF/VCD), componenta dinamica poate diferi in sarcina reala.

3. Performanta cycle-accurate (Verilator)

Simulare cu tb_wrapper_cycles.cpp pe top qnn_fpga_wrapper:

- **cycles_per_inference = 1047 cicluri**

3.1 Latenta

Cu **Fmax ≈ 198 MHz**:

- **Latenta ≈ 1047 / 198e6 ≈ 5.29 µs / inferenta**

3.2 Throughput

- **Throughput ≈ 198e6 / 1047 ≈ 189,112 inferente/s (~189k inf/s)**

4. Baseline software (Windows CPU/GPU)

Masurare cu PyTorch (dense 64→16 + ReLU):

- **CPU:** 9.379 µs/inferenta, **106,616 inf/s**
- **GPU RTX 3050 Ti (CUDA):** 52.456 µs/inferenta, **19,063.5 inf/s**

Nota. Pentru o operatie foarte mica (64→16), overhead-ul de lansare/sincronizare CUDA poate domina, de aceea GPU poate fi mai lent decat CPU.

5. Comparatie (FPGA vs CPU vs GPU)

5.1 Tabel sumar

Platforma	Latenta (µs)	Throughput (inf/s)
FPGA (Vivado + Verilator)	**5.29**	**189,112**
CPU (PyTorch)	9.379	106,616
GPU RTX 3050 Ti (PyTorch CUDA)	52.456	19,063.5

5.2 Raporturi de performanta

- **FPGA vs CPU:**
  - imbunatatire latenta: 9.379 / 5.29 ≈ **1.77**×
  - imbunatatire throughput: 189,112 / 106,616 ≈ **1.77**×
- **FPGA vs GPU:**
  - imbunatatire latenta: 52.456 / 5.29 ≈ **9.92**×
  - imbunatatire throughput: 189,112 / 19,063.5 ≈ **9.92**×

6. Concluzii

- Designul respecta constrangerile de timing, cu **WNS pozitiv** si frecventa estimata **Fmax ≈ 198 MHz**.
- Utilizarea resurselor este minima in configuratia evaluata: **61 LUT**, **78 FF**, **0 BRAM**, **0 DSP**.
- Simularea cycle-accurate arata **1047 cicluri/inferenta**, ceea ce corespunde la **~5.29 µs latenta** si **~189k inferente/s**.
- Comparativ cu baseline-ul PyTorch pe Windows, FPGA are **~1.77**× throughput mai mare decat CPU si **~9.92**× mai mare decat GPU pentru acest micro-benchmark.
- Power estimat: **0.071 W**, predominant static; consumul dinamic poate creste pentru activitate reala (SAIF).

7. Checklist livrabile

- util_impl.rpt (utilization)
- timing_impl.rpt (timing summary)
- power_vectorless.rpt (power report)
- Screenshot/log: cycles_per_inference=1047
- Screenshot: rezultate baseline CPU/GPU