# The Cost of Parallelism: A Measured Comparison of Sequential and Systolic INT8 GEMM on an FPGA

Boris Gordeev
Department of Electrical and Computer Engineering, Northeastern University

SC26 CHICAGO NOV 15–20

## Question & setup

**How much latency does spatial execution save, and what does it cost?**

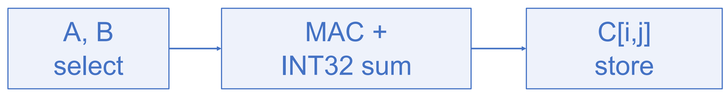
Controlled comparison: one reused MAC versus a conventional 4×4 output-stationary systolic array [1].

Same 4×4 workload, signed INT8/INT32 arithmetic and handshake. Separate core builds isolate area and timing [R, §3].

Cyclone 10 GX 10CX220YF780E5G
Quartus Prime Pro 26.1, 100 MHz constraint
Slow 900 mV, 0 °C, virtual matrix pins

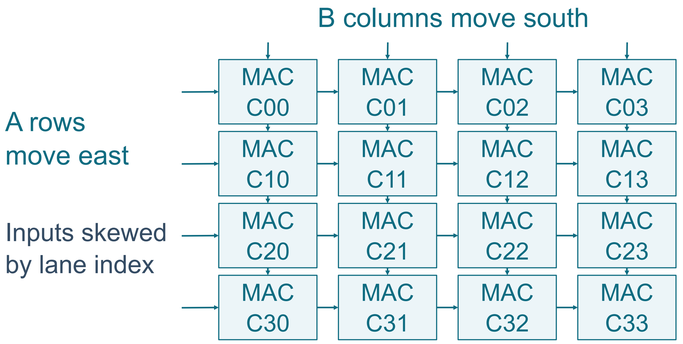
## Architectures

### Sequential: one shared MAC




**81 cycles = 1 init + 64 MAC + 16 store**

### Systolic: sixteen PEs




**11 cycles = 1 load + 10 feed**

Each PE retains one INT32 sum. All 16 results are exposed directly, without a store phase [R, §4].

## Results & validation

### 7.4× cycle speedup with 16× DSPs and 1.62× ALMs

**81 to 11 cycles for one 4×4 product**

| Metric | Sequential | Systolic |
|---|---|---|
| Latency (RTL) | 81 cycles | 11 cycles |
| DSP blocks | 1 | 16 |
| ALMs | 506 | 822 |
| Restricted Fmax | 162.6 MHz | 165.8 MHz |

Resources and restricted Fmax are from post-fit reports, with one build per engine [R, §6].

### Verification

RTL simulation reports 81 cycles for the sequential engine and 11 cycles for the systolic engine. Both engines passed fixed-vector FPGA self-tests.
A Signal Tap capture confirmed the systolic engine's 11-cycle busy window [R, Fig. 4 and Table 3].

Python reference verification with self-checking RTL tests covered 5 sequential, 5 systolic, and 4 engine-agreement checks.

### Calculated time ratio: 7.5×

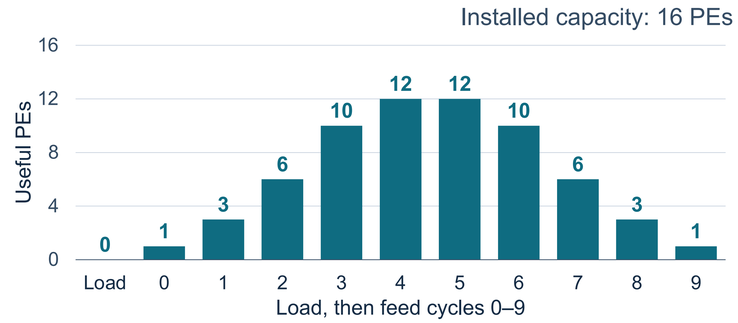
Cycles/Fmax: 498.2 ns sequential vs 66.3 ns systolic.
Core-only estimates, not measured board times.
External transfers excluded.

**Code & contact**

github.com/bgordeev/int8-systolic-matmul-fpga
gordeev.b@northeastern.edu

## Why the array is underutilized

Useful PE occupancy, derived from the documented skew schedule [R, §4.2]. Bars count useful MACs, not electrical switching.




**64 useful MACs. Peak: 12 of 16 PEs.**

### 36.4% useful utilization

64 / (16 × 11) for the whole operation
40% during the ten feed cycles

Baseline utilization is 64/81 = 79.0%.
16 × [(64/176)/(64/81)] ≈ 7.36× speedup

## Conclusions

The systolic engine achieves 36.4% useful MAC-slot utilization for this isolated 4×4 operation. This describes useful arithmetic occupancy. Electrical switching, power savings, and clock gating were not measured.

Evidence is limited to one device, one matrix size, and one build per engine.

Next: sweep sizes and placement seeds, vary workload size separately, and measure energy.

**References**

[R] B. Gordeev. The Cost of Parallelism. Report, 2026.
[1] H. T. Kung. Why Systolic Architectures? Computer, 1982. doi:10.1109/MC.1982.1653825