

Sie haben völlig recht mit der Annahme, dass für die Inferenz (und zunehmend auch für das Training) von neuronalen Netzen **FP32 und FP64 völlig überdimensioniert** sind. Genau dort liegt der Hebel: Weniger Präzision bedeutet kleinere Rechenwerke, weniger Speicherbandbreite und damit massive Geschwindigkeitsvorteile.

Tatsächlich tun die FPGA-Hersteller genau das, was Sie vorschlagen – und sie gehen sogar noch weiter. Die Antwort, warum das FPGAs trotzdem nicht automatisch zum Marktführer macht, liegt in einem **Missverständnis über moderne GPUs** und der harten Realität des Siliziums.

Hier ist die Erklärung, warum die Idee exzellent ist, aber GPUs das Rennen aktuell trotzdem gewinnen:

1. Die Hersteller bauen diese Kerne bereits (und zwar massiv)

Wie wir weiter oben bei den *AMD Versal (AI Engines)* und *Altera Agilex (AI Tensor Blocks)* gesehen haben, verbauen die Hersteller heute **festе, hochoptimierte INT8- und INT4-Tensor-Kerne** direkt auf dem FPGA-Silizium.

- Diese Blöcke machen im Grunde genau das, was Sie vorschlagen: Sie ignorieren FP32/64 und sind reine, fest verdrahtete Rechenmaschinen für extrem schnelle Ganzzahl-Matrix-Multiplikationen (INT8).
- In reinen Hardwaredisziplinen wie der **Latenz** (wie schnell reagiert das Netz auf ein einzelnes Bild/Signal) schlagen diese FPGAs jede GPU, da die Daten nicht erst durch komplexe Grafikkarten-Treiber und Betriebssystem-Schichten wandern müssen.

2. Das "Problem": GPUs haben das längst auch getan

Der Vorteil von GPUs bei FP32/64 gilt für die wissenschaftliche Forschung (z. B. Klimasimulationen). Für KI hat Nvidia jedoch vor Jahren eine radikale Kehrtwende vollzogen.

- Seit der Volta-Architektur (2017) und extrem ausgeprägt in aktuellen Generationen (wie Hopper H100 oder Blackwell) besitzen GPUs sogenannte **Tensor Cores**.
- Diese Tensor Cores in GPUs sind **keine** FP64-Einheiten. Sie sind hochspezialisierte, fest verdrahtete Beschleuniger im Grafikchip, die explizit für **INT8, INT4, FP8 und FP16/BF16** gebaut wurden.
- Da Nvidia diese Chips in gigantischen Millionenstückzahlen für Rechenzentren fertigt, können sie viel mehr dieser Tensor-Kerne auf eine GPU packen, als es auf einem teuren, flexiblen FPGA Platz hat.

3. Warum die GPU trotz INT8-FPGA-Kernen gewinnt

Wenn also beide (GPU und FPGA) spezialisierte INT8-Kerne haben, warum dominiert die GPU?

Kriterium	GPU (mit Tensor Cores)	FPGA (mit Tensor-Blöcken)
Der Speicher-Flaschenhals	Nutzt extrem schnellen HBM3e-Speicher mit mehreren Terabytes/Sekunde Bandbreite. Die Tensor-Kerne verhungern nie nach Daten.	Speicheranbindung oft langsamer (außer bei extrem teuren High-End-Modellen). Die Rechenkerne müssen oft auf Daten warten.
Software-Ökosystem	Der Hauptgrund: Ein Klick in PyTorch/TensorFlow genügt (model.to('cuda')), um das Modell auf INT8 zu quantisieren und sofort zu starten.	Das Modell muss durch den Compiler gejagt, die Hardware-Netzliste generiert und das Routing berechnet werden. Das dauert Stunden und erfordert Hardware-Wissen.
Änderungsgeschwindigkeit	Erscheint eine neue Netzwerk-Architektur (wie z. B. der Wechsel von CNNs zu Transformern/LLMs), wird einfach neuer Software-Code auf derselben GPU ausgeführt.	Bei radikal neuen KI-Architekturen muss oft das komplette Hardware-Layout des FPGAs neu designet und kompiliert werden.

Fazit

Ihre Intuition ist absolut goldrichtig: **INT8 ist der Standard für effiziente KI**, und FPGAs haben genau dafür eigene Tensor-Kerne bekommen.

FPGAs gewinnen diesen Kampf überall dort, wo **Echtzeit und geringe Latenz** lebenswichtig sind (z. B. wenn ein autonomes Fahrzeug ein Hindernis in unter einer Millisekunde erkennen muss). Wenn es aber darum geht, riesige Datenmengen mit maximalem Durchsatz im Rechenzentrum zu verarbeiten, zieht die GPU vorbei – nicht weil sie FP64 nutzt, sondern weil ihre eigenen INT8-Tensor-Kerne von einem unschlagbaren Software- und Speicher-Ökosystem gefüttert werden.

Möchten Sie tiefer verstehen, wie eine **INT8-Quantisierung** eines neuronalen Netzes funktioniert (also wie man ein FP32-Modell ohne großen Genauigkeitsverlust in INT8 umwandelt), oder interessiert Sie ein konkreter Vergleich der **Latenzzeiten** zwischen GPU und FPGA?