

FPGA-GPU Codesign

Heutige heterogene Computersysteme kombinieren CPUs, GPUs und FPGAs mit unterschiedlichen Architekturen und Nutzungsmöglichkeiten. GPUs mit immenser Parallelisierung sind bestens für die Echtzeit-Video- und Signalverarbeitung geeignet. Andererseits sind FPGAs ideal für die Erfassung und Vorverarbeitung mehrerer Videostreams. Daher ist die Aufteilung der Rechenaufgaben zwischen diesen Recheneinheiten und die Gestaltung der Datenpfade zwischen ihnen von größter Bedeutung. Ein enges und effizientes Zusammenspiel der einzelnen Komponenten ist daher unerlässlich.

Mit PCI-Express bieten moderne Computer ein schnelles internes Datenetzwerk zwischen CPU, GPU und FPGA. Da die Kommunikation jedoch in der Regel über die CPU erfolgt, wird diese Kommunikation zum Engpass in Anwendungen. Diese Arbeit vergleicht zwei verschiedene Implementierungen für ein effizientes FPGA-GPU Co-Design ohne CPU-Engpass.

Mit der ersten Implementierung namens XDMA richtet der Host jeden Datentransfer ein. Die andere Implementierung, genannt FDMA, wird einmalig vom Host initialisiert und benötigt keine weitere Host-Interaktion. Die XDMA-Implementierung nutzt bis zu 85% der PCI-Express-Bandbreite mit einem Übertragungsjitter von 3,4 ms, und die FDMA-Implementierung erreicht 72% mit einem Übertragungsjitter von 270 μ s. Ohne eine grafische Desktop-Umgebung des Linux-Hosts sank der Übertragungsjitter der FDMA-Implementierung auf 4 μ s.



Diplomand/in
Philipp Huber

Dozent/in
Hans-Joachim Gelke

Datenfluss zwischen der GPU und dem FPGA mit dem FDMA Konzept.

