Zurich University
of Applied Sciences

School of
Engineering

InES Institute of
Embedded Systems

# FPGA-GPU Co-Design

Today's heterogeneous computer systems combine CPUs, GPUs, and FPGAs with different architectures and usabilities. GPUs with immense parallelization are best fitted for real-time video and signal processing. On the other hand, FPGAs are ideal for capturing and pre-processing multiple video streams. Therefore, partitioning computational tasks between these computational units and designing the data-paths between them are of most importance. Close and efficient interaction of the individual components is, therefore, indispensable.

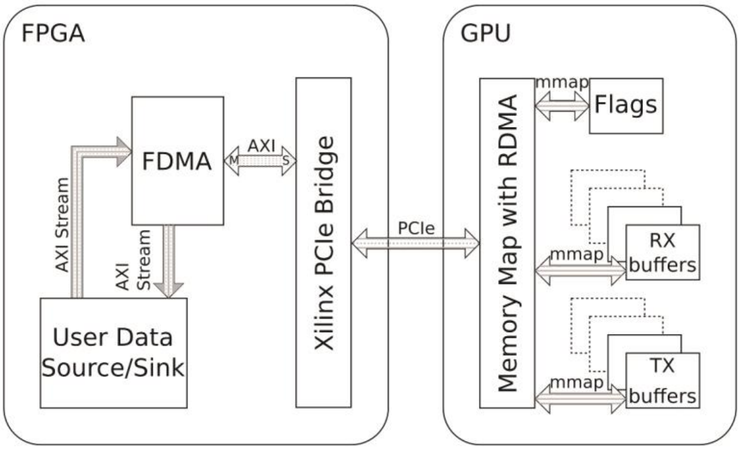
With PCI-Express, modern computers provide a high-speed internal data network between CPU, GPU, and FPGA. However, since communication usually takes place via the CPU, this communication becomes the bottleneck in applications. This thesis compares two different implementations for an efficient FPGA-GPU co-design without CPU bottleneck.

With the first implementation called XDMA, the host sets up each data transfer. The other implementation, called FDMA, is initialized once by the host and needs no further host interaction. The XDMA implementation uses up to 85% of the PCI-Express bandwidth with a transfer jitter of 3.4 ms, and the FDMA implementation reaches 72% with a transfer jitter of 270 µs. Without a graphical desktop environment of the Linux host, the transfer jitter of the FDMA implementation decreased to 4 µs.

Diplomand/in
Philipp Huber

Dozent/in
Hans-Joachim Gelke

Data flow between the GPU and the FPGA with the FDMA concept.


FPGA
FDMA
AXI
M
S
AXI Stream
AXI Stream
User Data Source/Sink
Xilinx PCIe Bridge
PCIe
GPU
Memory Map with RDMA
mmap
Flags
mmap
RX buffers
mmap
TX buffers