FPGA virtualization with accelerators overcommitment for Network Function Virtualization

Michele Paolino, Sébastien Pinneterre and Daniel Raho
Virtual Open Systems

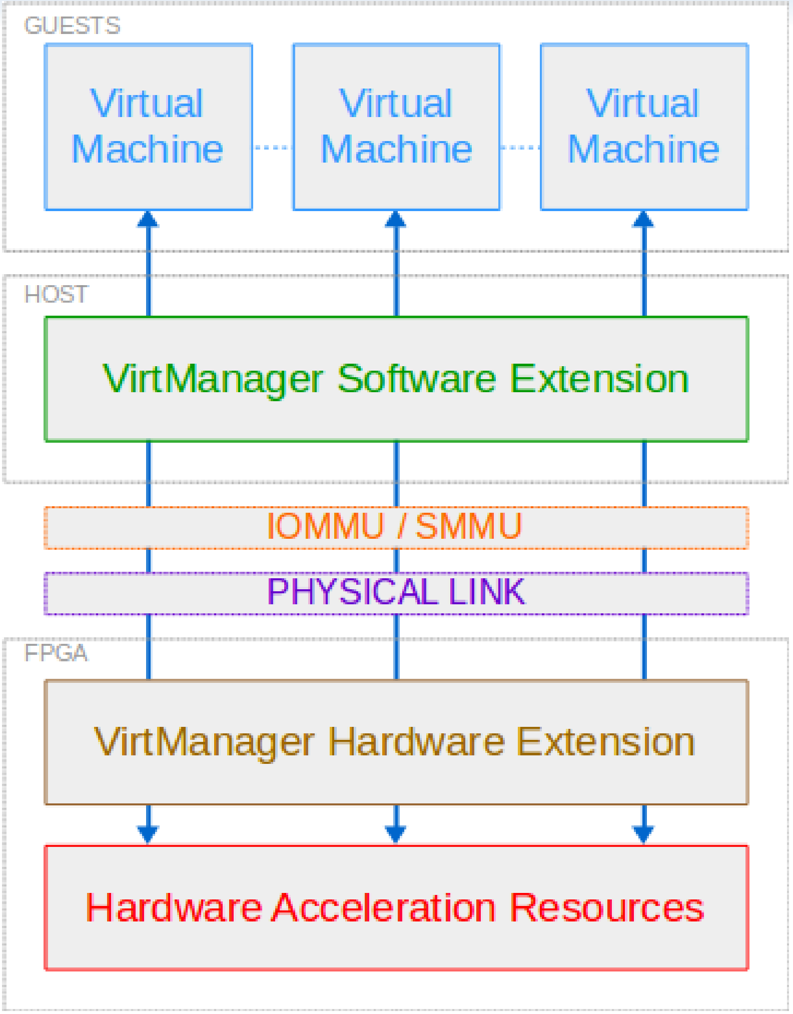
contact@virtualopensystems.com

Introduction

New lightweight virtualization techniques brought consolidation to its limits. This is particularly true in **NFV** where there is a need to run thousands of guests guaranteeing high performance and programmability.

VirtManager is an FPGA bitstream that enables accelerators consolidation by managing a specific context for each VM-accelerator connection:

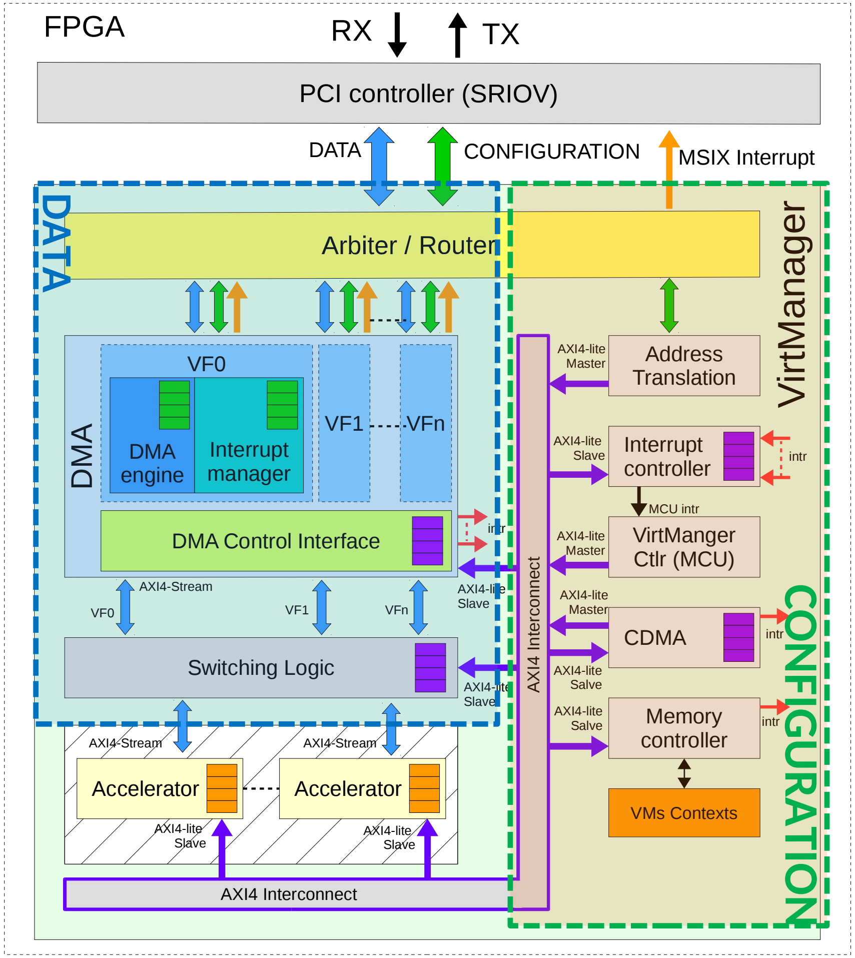
- ➢ Allocates an accelerator to multiple guests (needed for NFV microservices)
- ➢ Schedules accelerators deployment at run time based on QoS policies
- ➢ Presents a standard interface to software and accelerators, supporting existing software and accelerators


GUESTS
Virtual Machine
Virtual Machine
Virtual Machine
HOST
VirtManager Software Extension
IOMMU / SMMU
PHYSICAL LINK
FPGA
VirtManager Hardware Extension
Hardware Acceleration Resources

VirtManager

The purpose of this work is to present the VirtManager architecture and to assess the feasibility of its approach. VirtManager is composed by:

- An SR-IOV PCI Express controller
- A programmable Micro Control Unit (MCU) for scheduling, configuration, etc.
- A DMA engine for the datapath and a CDMA for the context switch operations
- Transfers arbitration component
- A context switch block which enables accelerator sharing among different VMs
- A Switch allowing the accelerator sharing among different VMs and the re-mapping between VMs and accelerators
- Standard interfaces to the hardware accelerators (AXI) and to the software (virtio)



Benchmark configuration

A set of benchmarks have been performed to evaluate feasibility of the context switch management. An interrupt signal has been used to trigger the context switch (CDMA configuration, data transfer to/from BRAM).

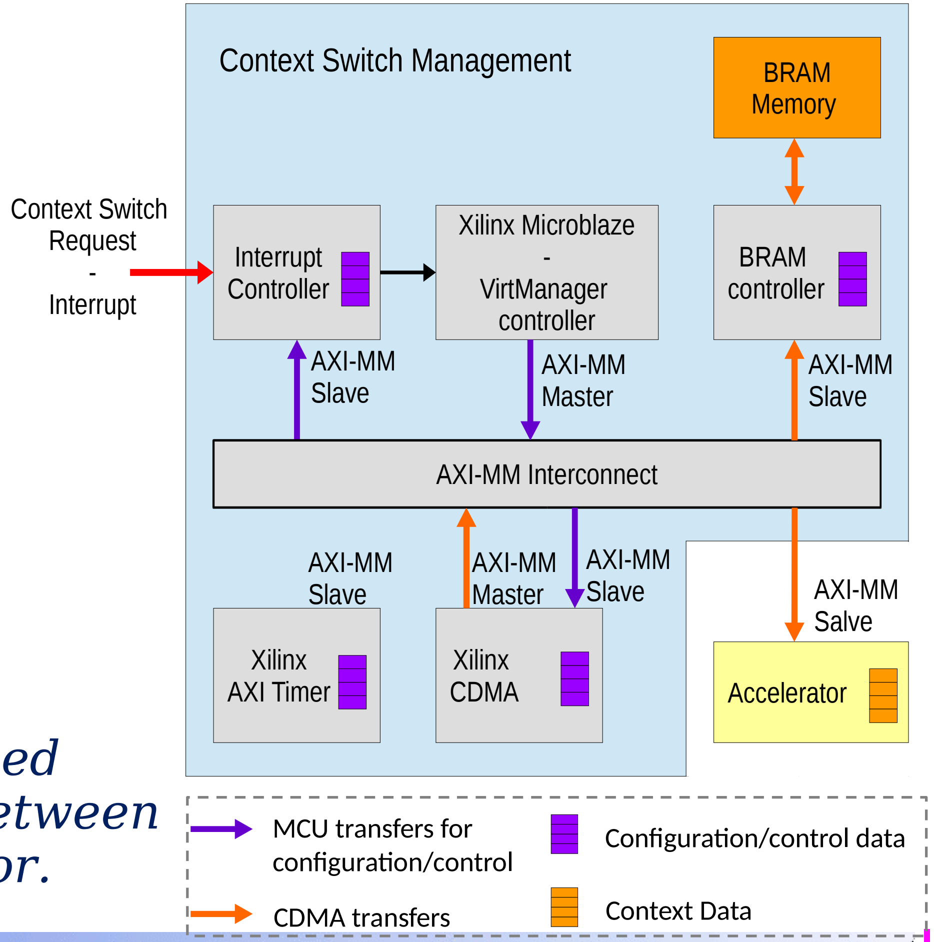
The prototype components are:

- MCU with context switch firmware
- AXI Timer, CDMA
- Bram controller/memory
- Interrupt Controller, AXI Interconnect

The performance has been measured with:

- Virtex 7 FPGA (XC7VH580T)
- Xilinx MicroBlaze (clocked at 100 MHz)
- Other components are clocked at 250 MHz

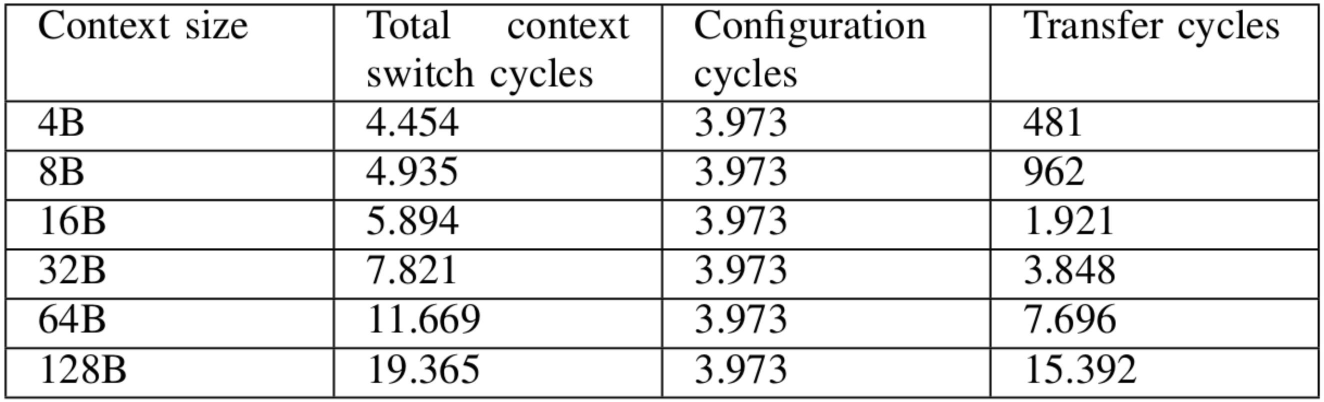
We define Context all the information needed (configuration, data, etc.) to support the link between a virtualized guest and a specific accelerator.



Benchmark results

The benchmark evaluation has been performed with different context size values and focusing on two operations:

- **Transfer**: transfers data to save and restore a context
- **Configuration**: configures central DMA to perform transfer



Context size	Total context switch cycles	Configuration cycles	Transfer cycles
4B	4.454	3.973	481
8B	4.935	3.973	962
16B	5.894	3.973	1.921
32B	7.821	3.973	3.848
64B	11.669	3.973	7.696
128B	19.365	3.973	15.392

With these results, and taking an FFT accelerator as a reference (262K computation cycles), we can claim a context switch overhead of ~2%.

We therefore consider this as proof of feasibility of the approach. Future work is an extension of the current prototype and a more extensive benchmark including the VMs datapath.

Virtual Open Systems