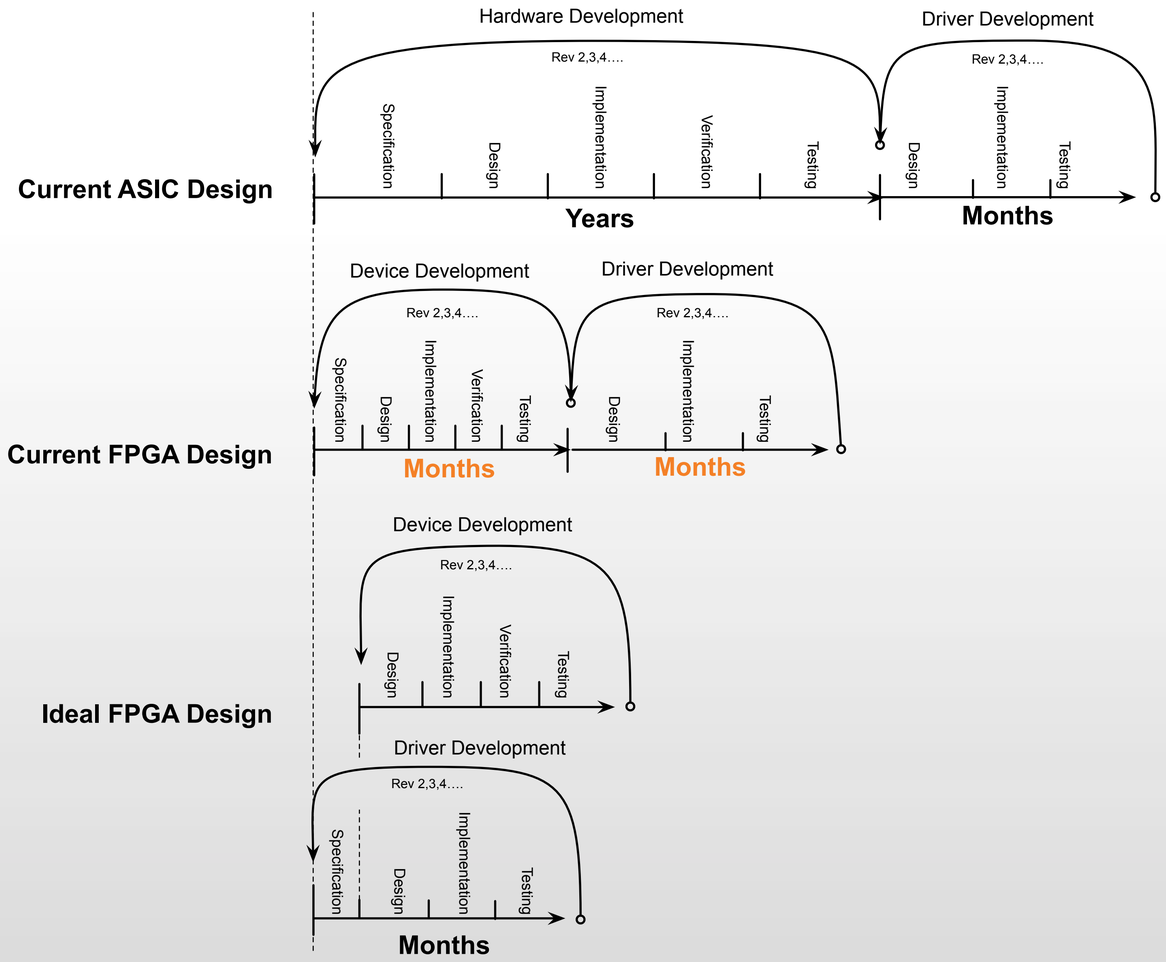
UNIVERSITY OF CAMBRIDGE

uvNIC: Rapid Prototyping NIC Device Drivers

M. P. Grosvenor



Network hardware isn't what it used to be

Traditional ASIC based network interface controllers (NICs) undergo minor hardware interface revisions over a timespan of years. FPGA based NICs can be completely reimplemented in months or even weeks.

Driver developers can't keep up

Driver development cannot seriously begin until hardware is available to test against, but driver development is expected to take place simultaneously with hardware development

What if driver developers could write the hardware?

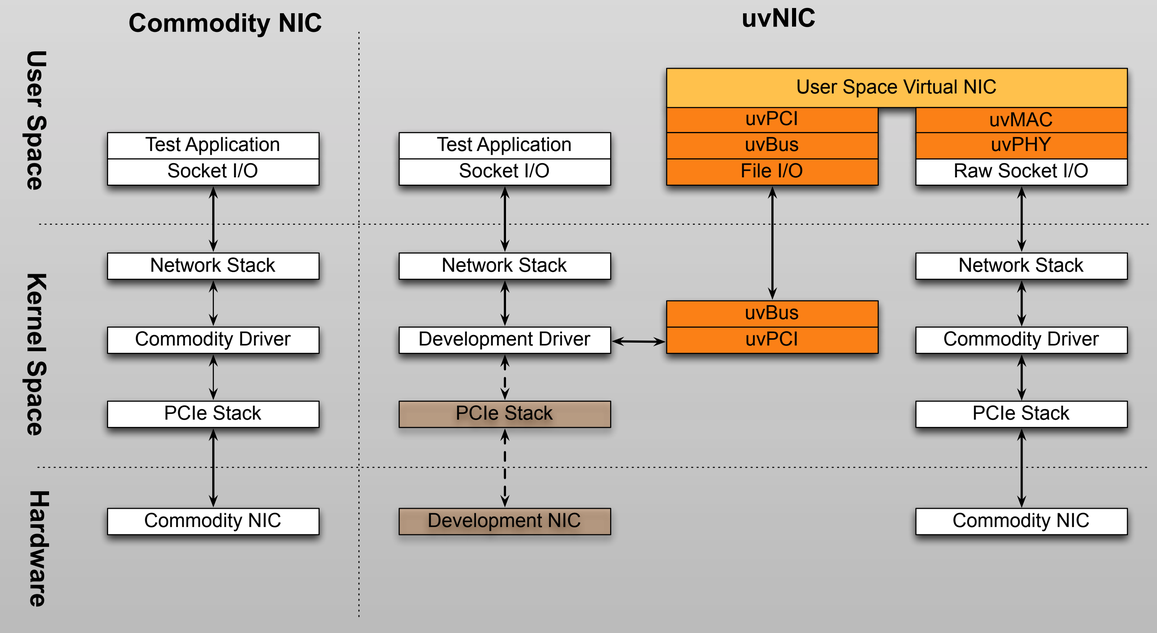
To the driver developer we could present a functional equivalent to a physical device. To the hardware designer we could present a fully functional model against which the HDL specifications could be tested and verified.

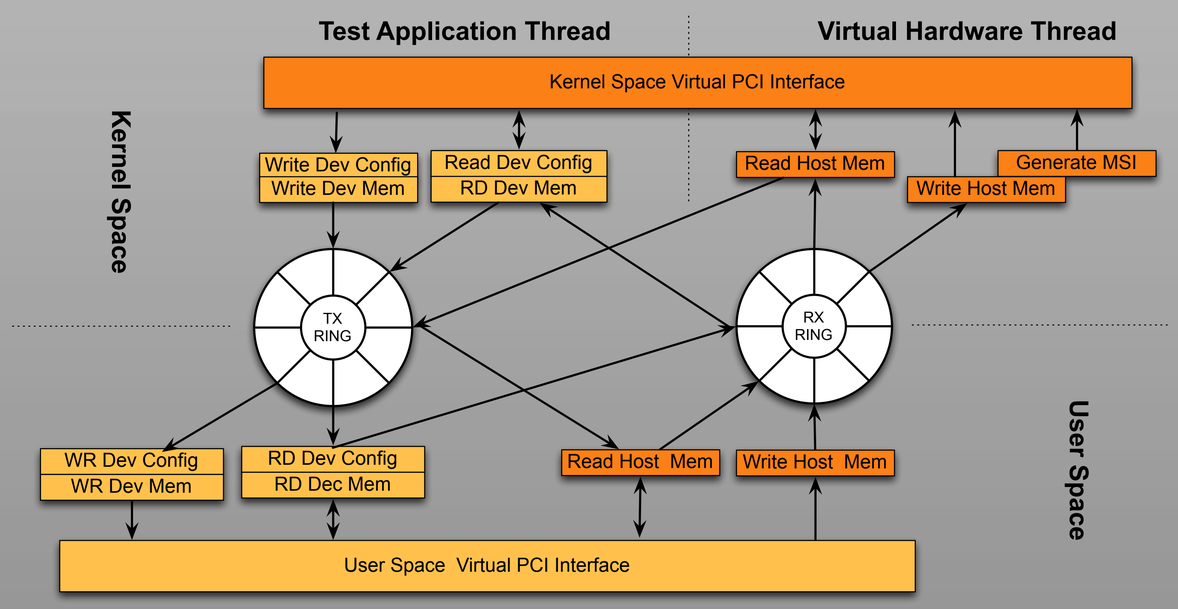
uvNIC: Making software look like hardware.

The user space virtual NIC is a standalone, userspace software application which is developed as a functional specification of a new NIC that is under development.

Key to uvNIC is the ability to augment an existing network interface card with new features and then write a functional device driver for the new virtual network interface.

The uvNIC device driver builds against a parallel implementation of the PCI kernel interface. Switching over to real hardware operation involves little more than a search/replace and a recompilation.





How do you make software look like hardware?

The user space virtual NIC is implemented on top of the user space virtual PCI (uvPCI) implementation, which itself is implemented on top of the user space virtual bus (uvBus) implementation.

The user space virtual bus makes the kernel dependent on user space in the same way that the kernel is dependent on hardware. This is kept safe by appropriate use of yield() and spinning timeouts.

By using a message passing transport layer, similar in design to hardware implementations of PCIe, important properties such as blocking reads and read/write/interrupt message ordering is maintained and consistent with reality.

Research Systems Group

srg@cambridge:~$

Approved for Public Release

This work was supported in part by the EPSRC INTERNET Project EP/H040536/1.

This research is sponsored by the Defense Advanced Research Projects Agency (DARPA) and the Air Force Research Laboratory (AFRL), under contract FA8750-11-C-0249. The views, opinions, and/or findings contained in this article/presentation are those of the author/presenter and should not be interpreted as representing the official views or policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the Department of Defense.

With thanks to Andrew W. Moore and Robert M. Watson

NetFPGA

EPSRC Pioneering research and skills