BlazePPS

A Packet Processing System Implemented in an FPGA and Linux

Valeh Amiri (vv2252)
Christopher Campbell (cc3769)
Sheng Qian (sq2168)
Yuanpei Zhang (yz2727)

Columbia University

May 14th, 2015

Overview

- Goals
- Design
  - Hardware
  - Software
- Results
  - Hardware
  - Software
- Lessons Learned

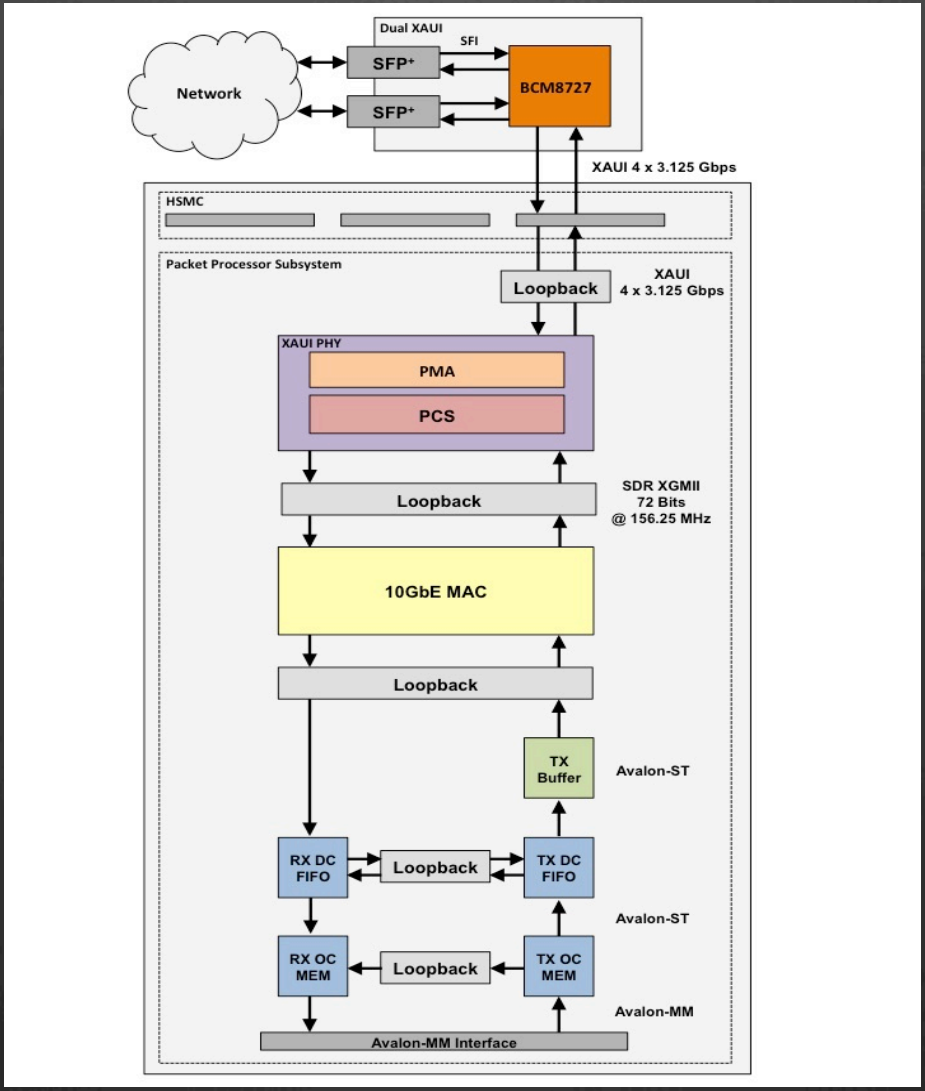
Goals

- Implement a 10 Gigabit Ethernet (10GbE) packet processing system in an Altera Cyclone V SoC FPGA
- Implement hardware and software
- Relatively inexpensive, flexible, and extensible system
- Easy to adapt to specific needs by modifying/adding modules
- Appealing for industries that wish to combine high-speed networking with hardware accelerated tasks

Design: Hardware

- Started with RocketBoards' Golden System Reference Design (GSRD)
  - Not useful, not necessary, and not working!
- Moved on to use Lab 3 design as basis
  - Much simpler and actually works!

Hardware Design



Design: Software

$$\begin{tikzcd}
\text{Memory Mapped Driver} \arrow[d] \\
\text{Network Driver w/ Software Loopback} \arrow[d] \\
\text{Network Driver}
\end{tikzcd}$$

Design: Software

- Driver registers as a platform device
- Allocates and registers a network/ethernet device
- Interface name given by kernel (ethx)
- MAC address and tx/rx FIFO addresses retrieved from Linux device tree
- Keeps interface statistics (can be seen using ifconfig)

Results: Hardware

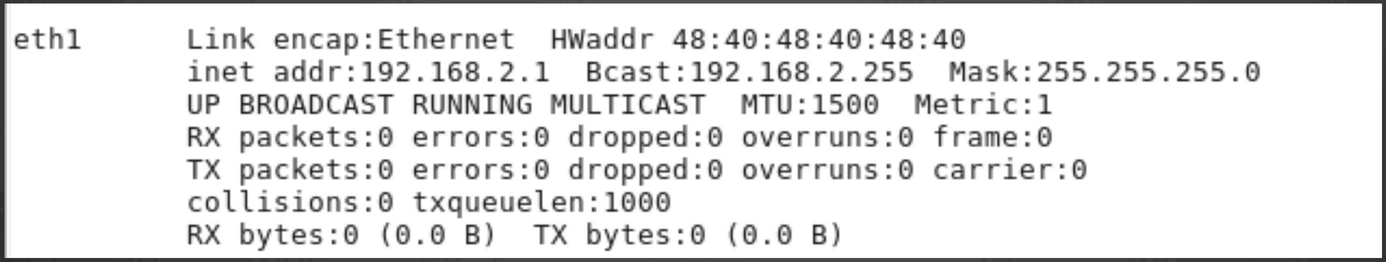
- System works with loopback paths that don’t involve MAC and on-chip FIFO at same time
- 10GbE MAC does not work when on-chip FIFOs are in the datapath
  - Extensive troubleshooting (tcl script testing, signal tapping, parameter modification, etc.)
  - No success
- System works front to back when using simplified MAC borrowed from previous project

Results: Software

- Transmit path works (must simulate end of transmission interrupt)
- Receive path works (must simulate receive interrupt)
- If we use hardware loopback and simulate transmit/receive interrupts we are able to send and receive a packet

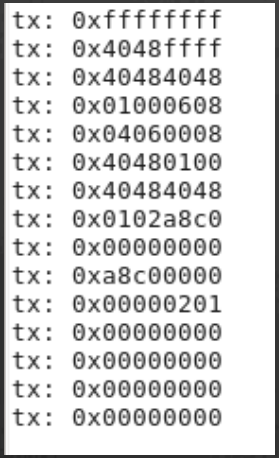
Results: Software

- **Loading the module and upping the interface**

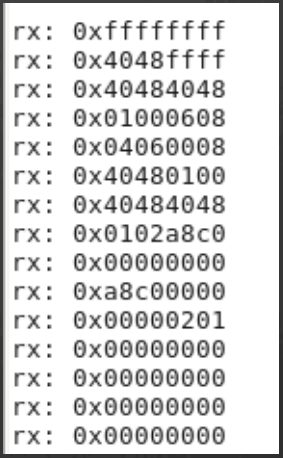


```
eth1      Link encap:Ethernet  HWaddr 48:40:48:40:48:40
          inet addr:192.168.2.1  Bcast:192.168.2.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

- **Transmitting and Receiving Packets**



```
tx: 0xffffffff
tx: 0x4048ffff
tx: 0x40484048
tx: 0x01000608
tx: 0x04060008
tx: 0x40480100
tx: 0x40484048
tx: 0x0102a8c0
tx: 0x00000000
tx: 0xa8c00000
tx: 0x00000201
tx: 0x00000000
tx: 0x00000000
tx: 0x00000000
tx: 0x00000000
```



```
rx: 0xffffffff
rx: 0x4048ffff
rx: 0x40484048
rx: 0x01000608
rx: 0x04060008
rx: 0x40480100
rx: 0x40484048
rx: 0x0102a8c0
rx: 0x00000000
rx: 0xa8c00000
rx: 0x00000201
rx: 0x00000000
rx: 0x00000000
rx: 0x00000000
rx: 0x00000000
```

Lessons Learned

- Have more clearly defined goals and stick with them
- It can be easier to do something from scratch rather than extending someone else’s work

Credits

Dr. Stephen Edwards
Professor David Lariviere
Bhargav Sethuram
John Chaiyasarikul
Yumeng Xu
Shuguan Yang
Jian Zhong