Extending/Optimizing the USRP/RFNOC Framework for Implementing Latency-Sensitive Radio Systems

Joshua Monson*, Zhongren Cao^, Pei Liu~, Travis Haroldsen*, Matthew French*

11/14/2018

*USC Information Sciences Institute Arlington, VA
^C3-COMM Systems Vienna, VA
~New York University

4676 Admiralty Way
Marina del Rey, CA

3811 N Fairfax Drive
Arlington, VA
USC Information Sciences Institute

- **Reconfigurable Computing Group @ USC/ISI**
  - Over 20 years performing cutting edge FPGA research
  - > 100 Journal and Conference publications
  - Focused on FPGA and ASIC, System-Level Design, Productivity, TRUST and Security
  - Custom ASIC/FPGA CAD Tool (TORC)

- **ISI: A Large, vibrant, path-breaking research Institute**
  - Part of USC’s Viterbi School of Engineering located in “Marina Tech Campus” (Marina del Rey) and in Arlington, VA
  - >$80M per year in funding from a diversified base of sponsors
  - ~300 people mostly research staff
  - Facilities to conduct ITAR, classified, and unclassified research
Overview

• Discuss Extensions/Optimizations to the UHD/RFNOC that allowed us to meet stringent \textit{latency requirements} of the transceiver

• Experiences, lessons learned, and development efforts to implement a broadband CSMA/CA based OFDM transceiver on an Ettus E310 USRP Software Defined Radio (SDR) platform.

• Supports link layer latency-cooperative transmissions
• Over 95% of casualties in Operations Enduring Freedom, Iraqi Freedom, and New Dawn occurred after operations transitioned from linear, conventional fights to the nonlinear, nonconventional stabilization phase

— A majority of missions during the nonconventional stabilization phase are carried out by dismount squads at the tactical edge

— Sharing situational awareness among soldiers is vital to mission successes

— Multicast plays an increasingly important role in edge networks

• Goal: Improve the throughput of wireless multicast and broadcast in dismounted squad networks in order to significantly enhance the situational awareness at the tactical edge

Sources:
COMBAT Innovations

- COMBAT system works to support and improve IP multicast schemes such as the SMF in RFC 6621
  - IP multicast to provide connections among clusters
  - Local mobility within clusters is handled by COMBAT in the link layer

- Higher Data Rate
- Transparent to IP Layer
- Responsive to Channel
- Efficient Channel Utilization
- Reduced Complexity
System Level Simulation

- Simulation Setup
  - Topology: Mobile distributed on a disc R=100 meter
  - Radio Propagation: Path loss + AWGN
  - Traffic: Multicast only from a random node.
  - Random Waypoint Model: Velocity (0.1-4 m/s), Pause duration (0-60 s).
COMBAT Objectives

• Demonstrate Improvements in Relay Throughput on Prototype OFDM Transceiver over Worst Link Scenario

• Requirements:
  – 10 MHz Bandwidth, 40 MSPS
  – Utilize CSMA/CA
COMBAT Latency Requirements

• Contention-free relay with PHY-assisted ACK/NACK

• CSMA/CA deadlines
  – Enable Compatibility with Existing Systems

• Throughput
  – RX-to-TX Latency is critical for throughput
Software-Defined Architecture

- USRPs are latency-insensitive peripherals
- Latency is low, but SDRs not intended to meet stringent latency CSMA/CA deadlines
Enabling Advances in SDR Architecture

Standard SDR Architecture

Host

Ethernet/USB Connection

FPGA

RF-Front

Block Diagram of E310

Embedded SDR Architecture

- Single Package
- Host/FPGA Latency Smaller
- Larger FPGA

Block Diagram of N210

ARM CORTEX A9

ADC

DAC

Decimation

Interpolation

Block Diagram of E310

ARM CORTEX A9

ADC

DAC

Decimation

Interpolation

Analog Filter Banks

Analog Filter Banks

Information Sciences Institute
Radio Frequency Network on Chip (RFNOC)

- Dynamically Programmable Network-on-Chip
- Provides a design entry-point into the FPGA
RFNOC Initial Configuration on E310
E310-RFNOC Base Design (FPGA Only)

NOC IF are ~30% Resource Utilization
E310 RFNOC Base Design (FPGA Only)

- ZYNQ FIFO
  - 16 Channel Host-to-RFNOC
  - 8 Channel DMA
- FIR FILTER
- FFT
- SIGNALGEN

*Required Extension/Update UHD
E310 RFNOC Base Design (FPGA Only)

**Freed 53% of the FPGA Resources!**

<table>
<thead>
<tr>
<th></th>
<th>LUT</th>
<th>FF</th>
<th>BRAM</th>
<th>DSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total ZYNQ 7020 Device</td>
<td>53,200</td>
<td>106,400</td>
<td>140</td>
<td>220</td>
</tr>
<tr>
<td>RFNOC (Initial)</td>
<td>41,247 (77%)</td>
<td>55,783 (52.4%)</td>
<td>116 (82.8%)</td>
<td>146 (66.3%)</td>
</tr>
<tr>
<td>RFNOC baseline (RFNOC/Radio)</td>
<td>12,546 (23.5%)</td>
<td>15,840 (14.8%)</td>
<td>26 (18.6%)</td>
<td>0 (0%)</td>
</tr>
<tr>
<td>COMBAT optimized</td>
<td>45,319 (85.1%)</td>
<td>52,540 (47%)</td>
<td>104.5 (74.6%)</td>
<td>120 (52%)</td>
</tr>
</tbody>
</table>
• Packet Buffering into RFNOC is Primary Cause of Delay.
• Dependent on Programmable Packet Size.
• Static Delay through RFNOC .625 us

RFNOC CLOCK = 50 MHz (400 MB/Sec)
CE CLOCK = 40 MHz (40 MSPS, 160 MB/Sec)
RFNOC Latency

Minimum RX –to-TX Delay

<table>
<thead>
<tr>
<th>Component</th>
<th>Delay (µs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFNOC</td>
<td>3.0</td>
</tr>
<tr>
<td>ADC/DAC IF</td>
<td>~250 ns</td>
</tr>
<tr>
<td>RF-Frontend</td>
<td>5.0</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>8.25</strong></td>
</tr>
</tbody>
</table>

RFNOC CLOCK = 50 MHz (400 MB/Sec)
CE CLOCK = 40 MHz (40 MSPS, 160 MB/Sec)
Low-Latency RFNOC Extension

Low-Latency RFNOC Block Benefits:
- Minimum Latency Limited by RF configuration and a couple cycles
- Selectable – Select LL for TX, RX, or TX/RX
- Maintains Compatibility with RFNOC/UHD; however needs additional work to support GNURadio

Minimum RX –to-TX Delay

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>RFNOC</strong></td>
<td>3.0 us</td>
</tr>
<tr>
<td><strong>ADC/DAC IF</strong></td>
<td>~250 ns</td>
</tr>
<tr>
<td><strong>RF-Frontend</strong></td>
<td>5.0 us</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>5.25 us</td>
</tr>
</tbody>
</table>

RFNOC CLOCK = 50 MHz (400 MB/Sec)
CE CLOCK = 40 MHz (40 MSPS, 160 MB/Sec)
Other Mods/Extensions to E310/UHD

• **Updates to REPO**
  – Added Block Diagram Build Flow
  – Construct Vivado GUI Project for Block Diagrams
  – Smoother Integration Vivado IP

• **Exposed AD9361 Control**
  – Exposed SPI Interface to Write AD9361 Control

• **Debugging**
  – Built Custom Cable and Implemented Virtual JTAG
  – Integrated Virtual JTAG Server/Driver
OFDM Transceiver Architecture

1. **Upper MAC**
   - a) Wraps payloads in packets
   - b) Manages COMBAT protocols
   - c) Moves packets to and from packet buffers

2. **Lower MAC**
   - a) Configures and manages RX and TX
   - b) Handles latency sensitive protocol (e.g., relay forwarding, etc.)
   - c) Inform Upper MAC of received packets.

3. **Packet Buffers**
   - a) TX: Hold packets that have been staged from transmission
   - b) RX: Hold packets that have been received and decoded

4. **OFDM RX/TX**
   - a) TX: Transform bits to OFDM baseband samples.
   - b) RX: Transforms OFDM baseband samples to bits

---

**ZYNQ 7020 FPGA**

- **Performance Statistics**
  - 8 Rate Settings (3-27 Mb/s)
  - 40 MSPS, 10 MHz BW

**Control BUS**

---

**Family:**
- ARM (Hard IP)
- MicroBlaze

**Interface:**
- RFNOC
- AD9361 (ADC/DAC IF)
- Mailbox

---

DISTRIBUTION A. Approved for public release: distribution unlimited.
Latency Analysis

ZYNQ 7020 FPGA

Upper MAC
ARM (Hard IP)
RFNOC

OFDM TX
OFDM RX
AD9361 IF

Type Buffers

Lower MAC
MircoBlaze

DISTRIBUTION A. Approved for public release: distribution unlimited.
Latency Analysis

ZYNQ 7020 FPGA

ARM (Hard IP)

Upper MAC

RFNOC

OFDM TX

17

.2

Packet Buffers

Lower MAC

MircoBlaze

Mailbox

DISTRIBUTION A. Approved for public release: distribution unlimited.
Latency Analysis

ZYNQ 7020 FPGA

ARM (Hard IP)

Upper MAC

RFNOC

OFDM TX

AD9361 IF

TX

RX

Packet Buffers

Lower MAC

MircoBlaze

Mailbox

17

.2

4.1

2.5

AD9361
Latency Analysis

ZYNQ 7020 FPGA

Upper MAC
ARM (Hard IP)

RFNOC

AD9361 IF

TX

RX

Packet Buffers

Lower MAC
MircoBlaze

Mailbox

1

17

.2

4.1

2.5

DISTRIBUTION A. Approved for public release: distribution unlimited.
Latency Analysis

29.3 us

ZYNQ 7020 FPGA

ARM (Hard IP)

Upper MAC

RFNOC

AD9361 IF

Packet Buffers

Lower MAC

MircoBlaze

DISTRIBUTION A: Approved for public release: distribution unlimited.
RFNOC Latency Measurements

- NOC Delays Primarily Related to RFNOC Packet Size & Buffering
- Opportunities to further optimize MAC Code can be optimized
- RX-to-TX Latency exceeds Requirement (34 us)!
- Best Case Relay of 10.98 Mb/S
Low-Latency IF Measurements

- Reduced RX-to-TX Latency by 50%!
- Optimizations:
  - RFNOC
    - Adjusted FC Buffers
    - Adjusted FC ACKs
  - MAC Code
    - Added –O3 (50%)
    - Adjusted Algorithms
    - Overlapped More relay processing with RX
    - Added B. Shifter, Mult

- Meets Latency Requirement
+4.7% Achievable Relay Throughput!
- 2.5% Packet loss due to TX Underrun
  - Likely due to Flow Control ACK
Low-Latency IF Measurements

OFDM Relay 18 Mb/s Rate

Relay Latency: 22.6 us
# RFNOC Review

<table>
<thead>
<tr>
<th>Benefits</th>
<th>Drawbacks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple FPGA Design Entry Point into Ettus Stack</td>
<td>Not Intended for ultra-low latency processing</td>
</tr>
<tr>
<td>Excellent for High-level Design Tools like GNURadio/RFNOC</td>
<td>No Abstraction for Low Latency Signals communicating between blocks (e.g. state-inputs and outputs)</td>
</tr>
<tr>
<td>Excellent for Modular Design</td>
<td></td>
</tr>
<tr>
<td>Flexible</td>
<td></td>
</tr>
<tr>
<td>Stock IP to Connect To RFNOC</td>
<td></td>
</tr>
</tbody>
</table>

- We will be releasing our [Low-Latency RFNOC Block](https://github.com/ISI-RCG) and [NOC Radio Extension](https://github.com/ISI-RCG) as open-source shortly on [https://github.com/ISI-RCG](https://github.com/ISI-RCG) (likely location)
- Recommended using these cores for ultra-low latency processing
Conclusion and Summary

• Implemented Latency-Sensitive CSMA/CA Radio System on USRP E310/E312

• Made Several Extensions and Improvements to Permit Latency Sensitivity
  – Created Low-Latency Radio Block and Low Latency RFNOC Block Shell
  – UHD Updates that Permitted RX->TX without Host
  – Enabled Vivado Block Diagram Build Flow
  – Implemented Lower MAC In Microblaze Processor
  – Enabled TX to Read from RX Packet Buffer
  – Transfer Packets to/from FPGA Rather than Samples

• Future Work:
  – Push Abstraction Levels Higher (Mapping to GNURadio)
Questions