Timing Analysis of LimeSDR

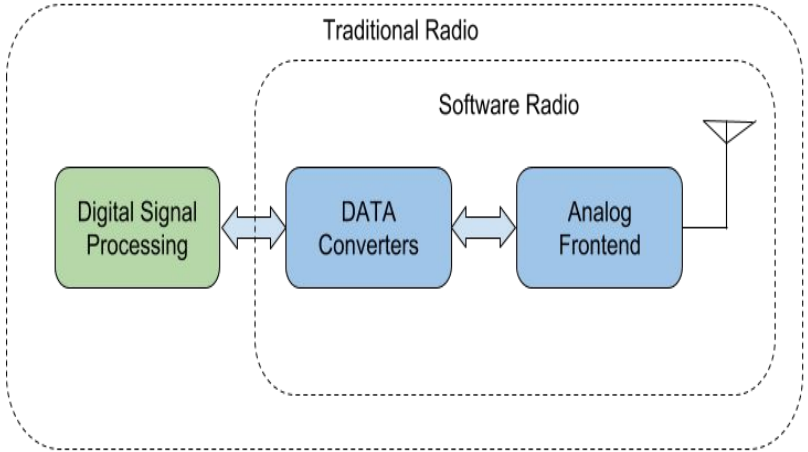
Saptarshi Hazra

Software Defined Radio

- Flexible Radio Communication System.
- Flexibility provided by implementing signal processing tasks on General Purpose Hardware.

Advantages:

- Possibility of future proof systems.
- Rapid evolvement of communication standards.



Motivation

- Possibility of technology agnostic radio-head.
- Design of Cognitive Radios.

Problem Area:

Challenges:

- Need for computational horsepower.
- Introduction of additional communication delays.
- Non-deterministic behaviour.

Impact:

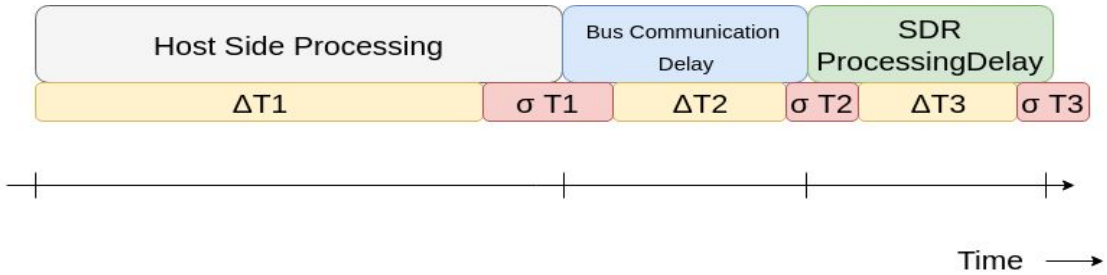
- Unable to implement time-sensitive operations of modern MAC.
- TDMA MAC protocols difficult to implement.

Research Question:

Q: Feasibility study of TDMA protocols on LimeSDR.

Q.1: What are the communication bottlenecks in LimeSDR implementation?

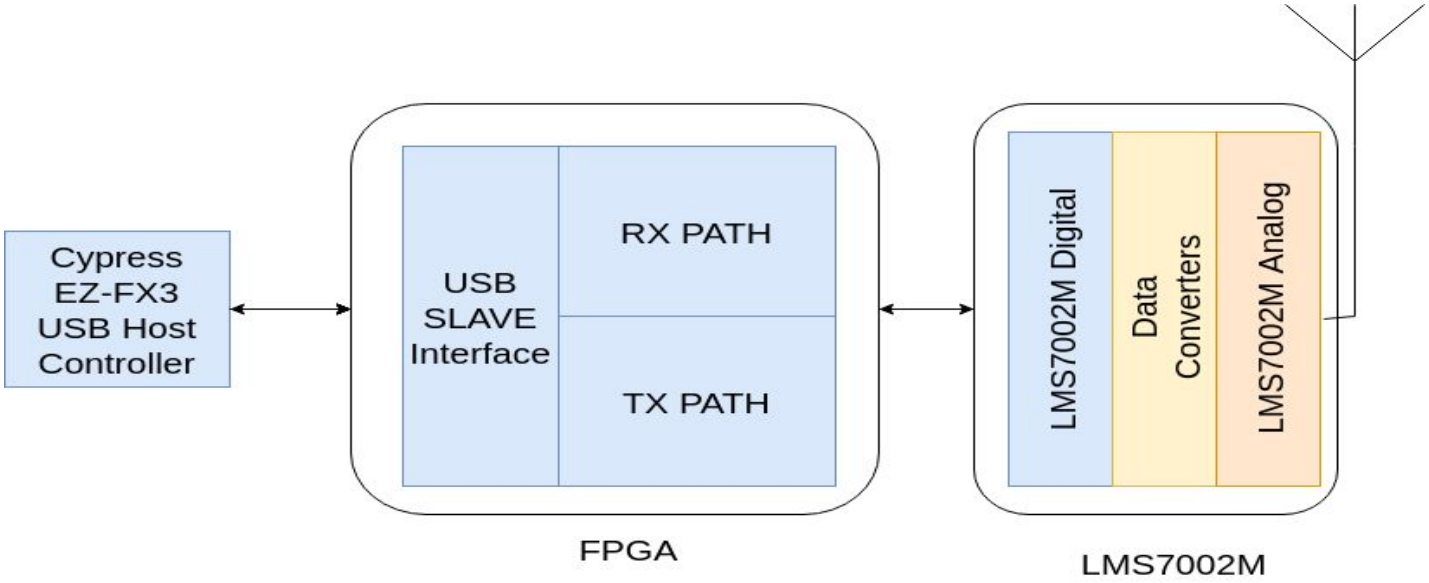
- **Primary Measurement Metric:** Jitter.
- **Secondary Measurement Metric:** Latency.



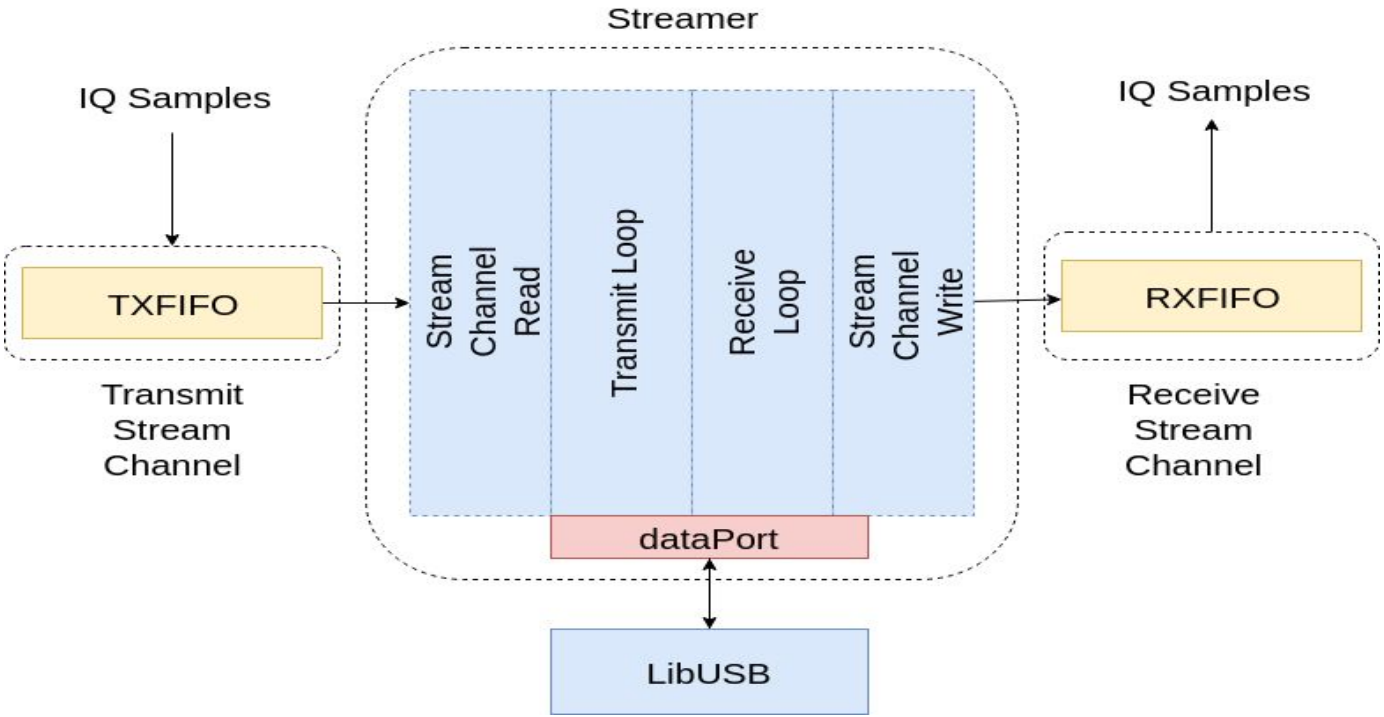
Overview of the Presentation.

- Introduction to LimeSDR.
- Brief Introduction to GNURadio.
- Method Explanation.
- Results.
- Analysis.

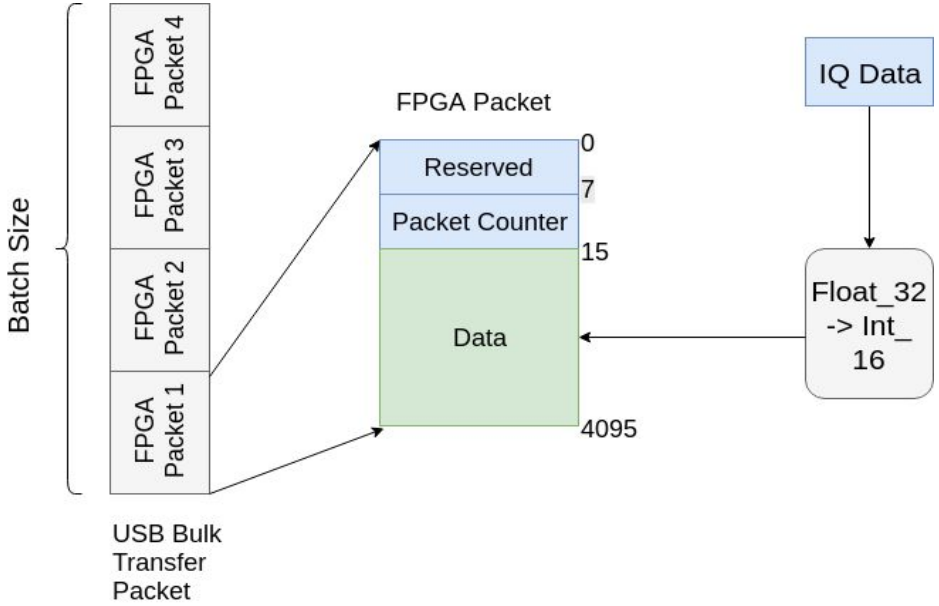
LimeSDR Hardware Architecture:



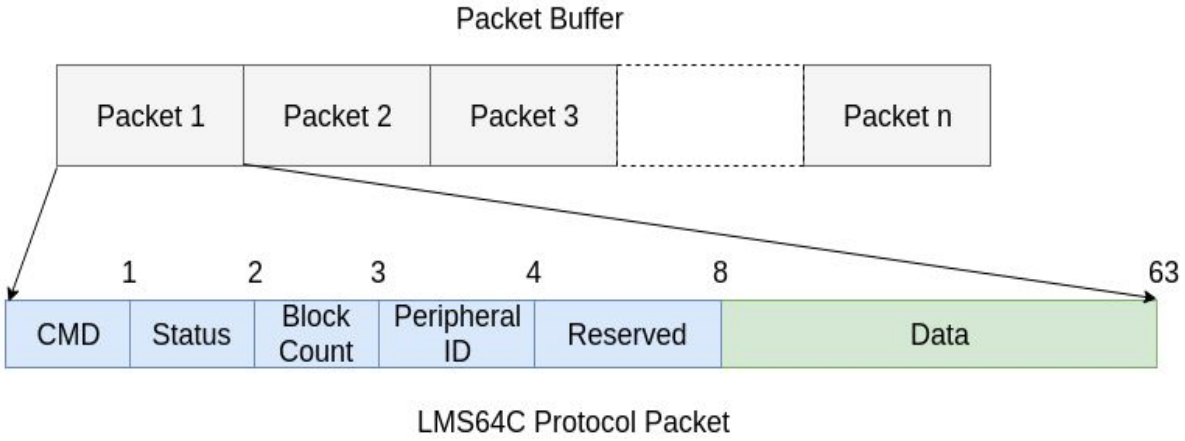
LimeSDR Software Architecture:



LimeSDR- Transmit Loop (Data Packets)

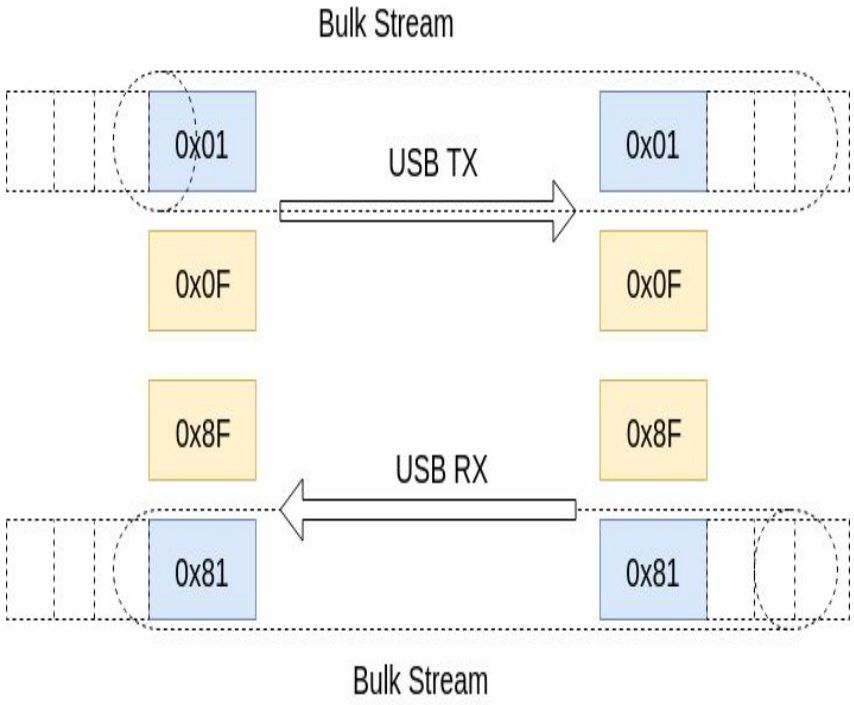


LimeSDR- Control Packets



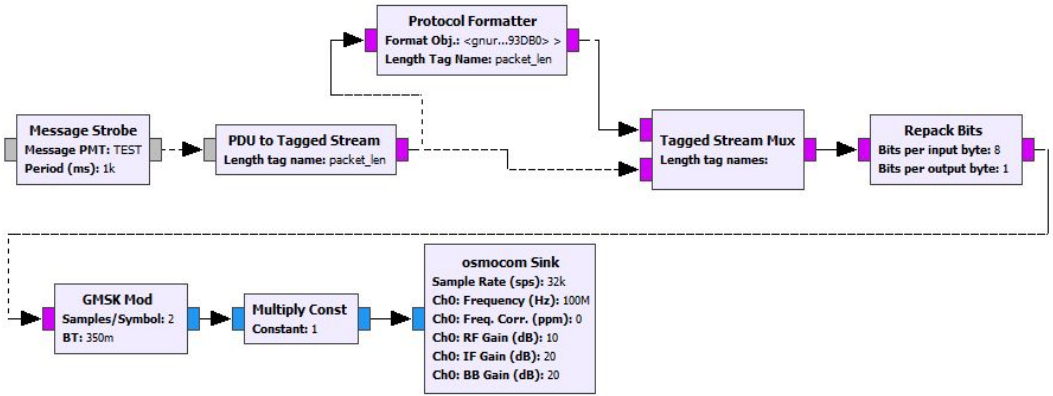
LimeSDR-USB Configuration:

EndPoint	Function
0x01	Stream Data Out
0x81	Stream Data In
0x0F	Control data Output
0x8F	Control Data Input



GNURadio

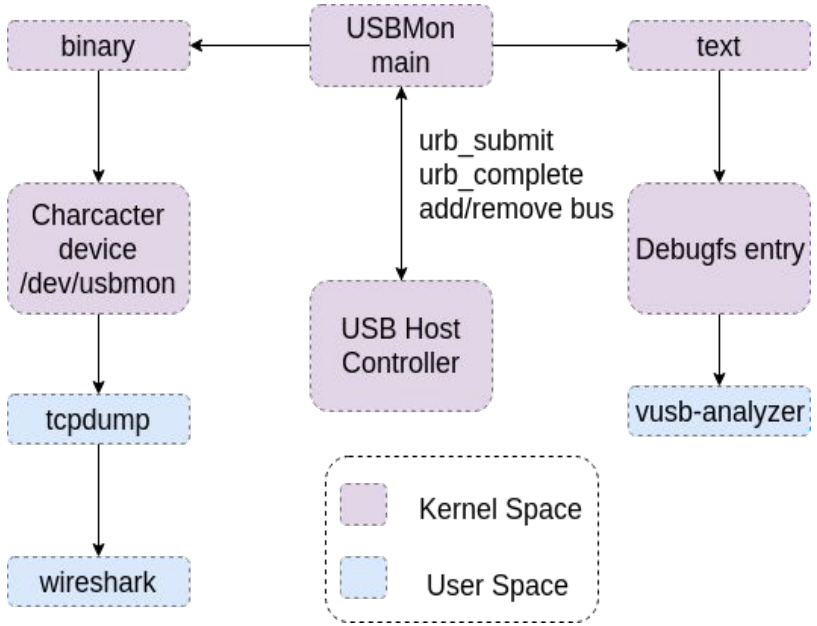
- Software development toolkit used to design processing blocks for Software Defined Radio.
- These blocks can then be combined together in the form of flowgraphs.



Method: Quantitative Analysis

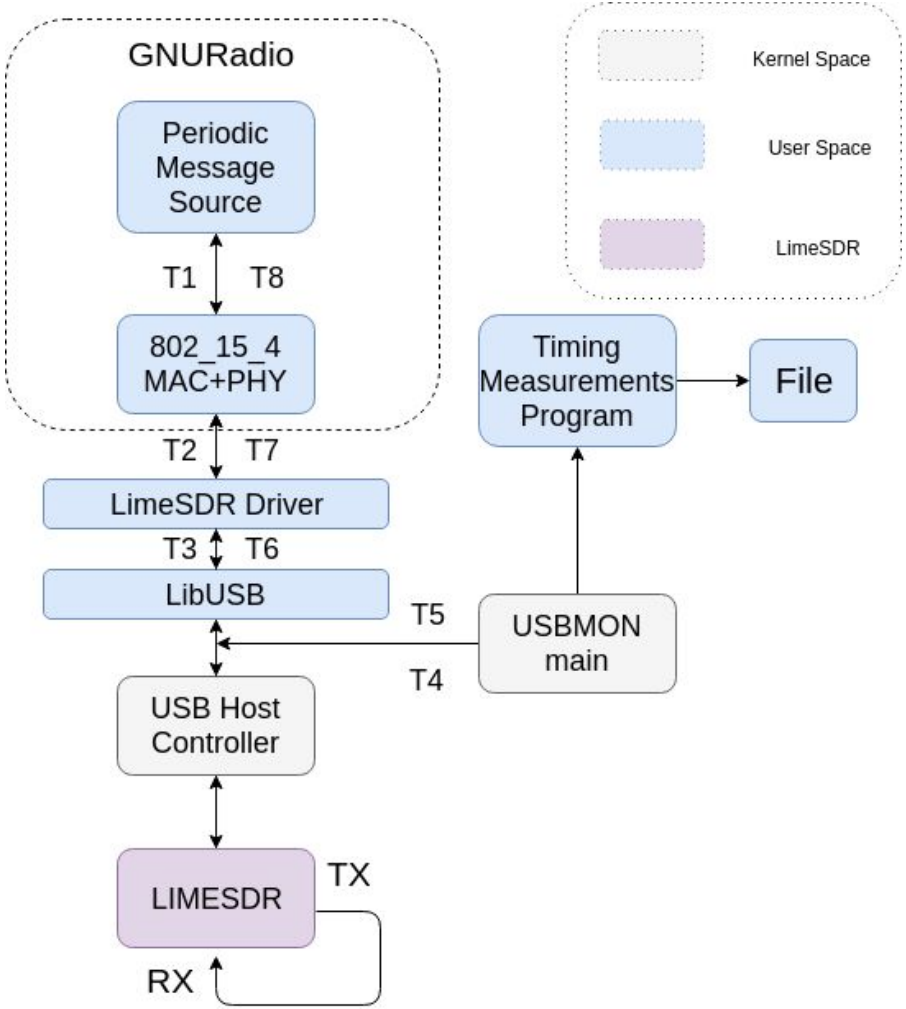
USBMon:

- Kernel utility to collect I/O traces on the USB Bus.
- Reports the requests made to and by the USB Host Controller.

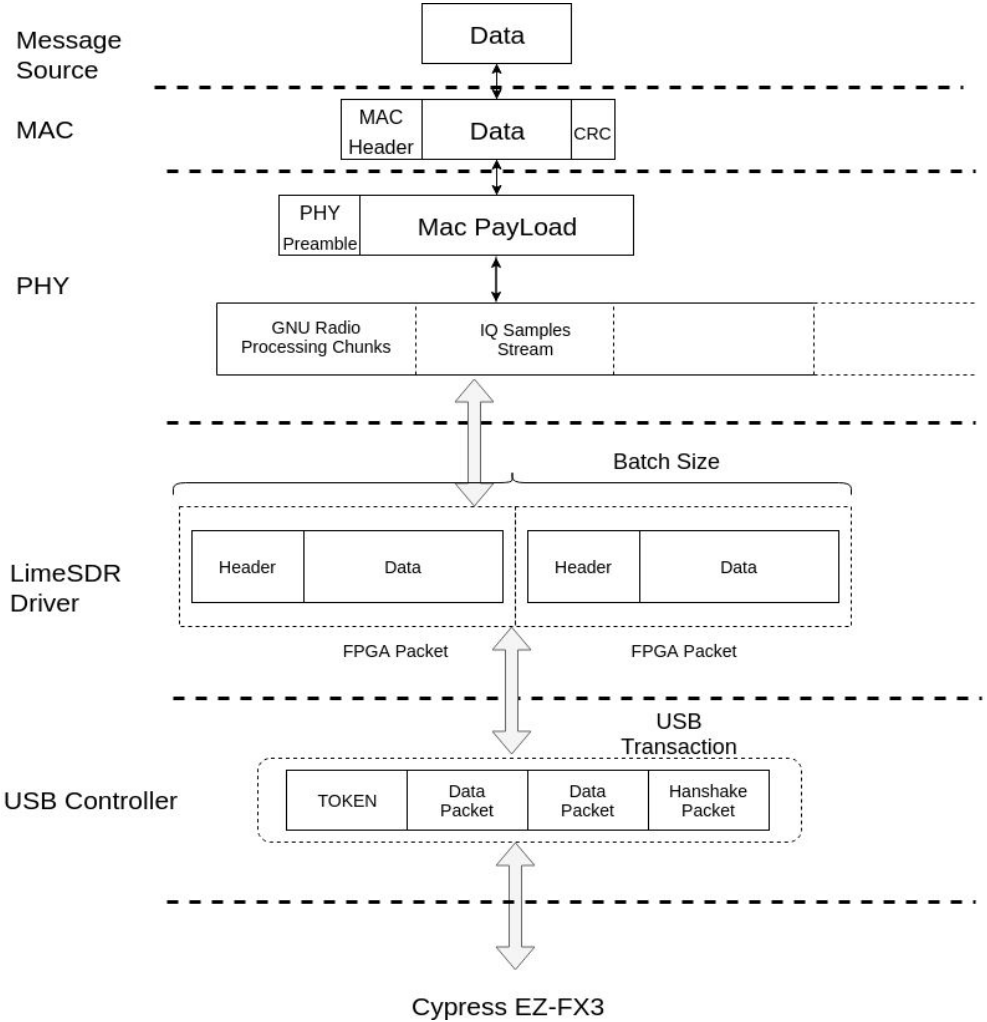


USBMon Architecture

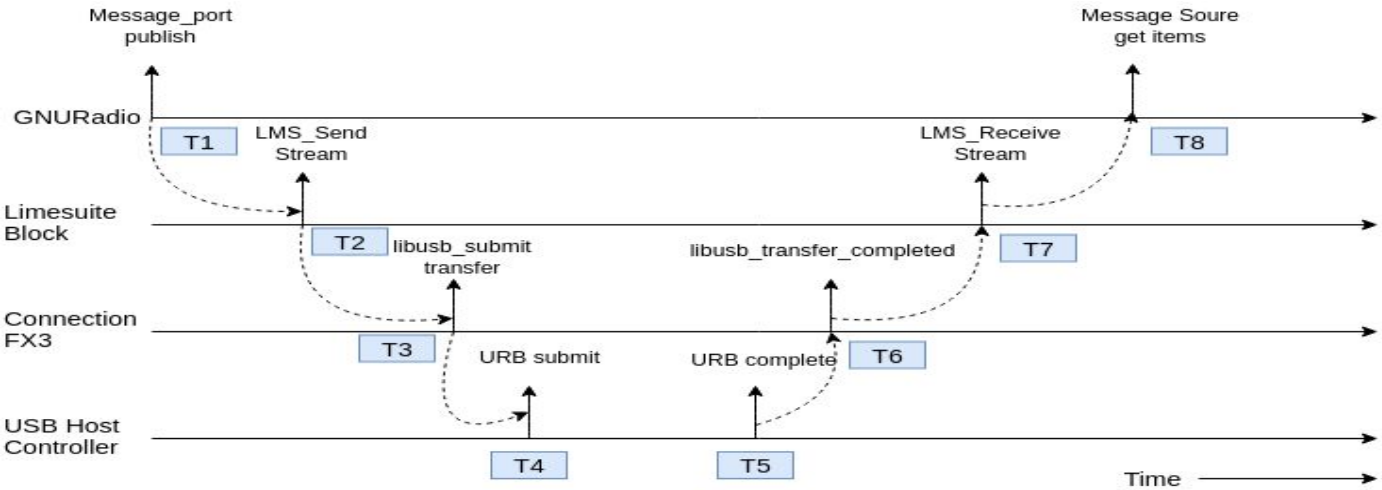
Measurement Setup



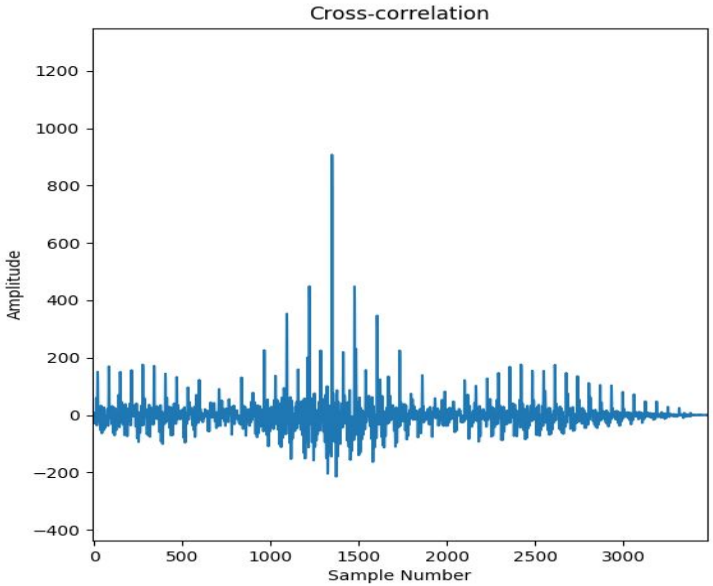
Dataflow

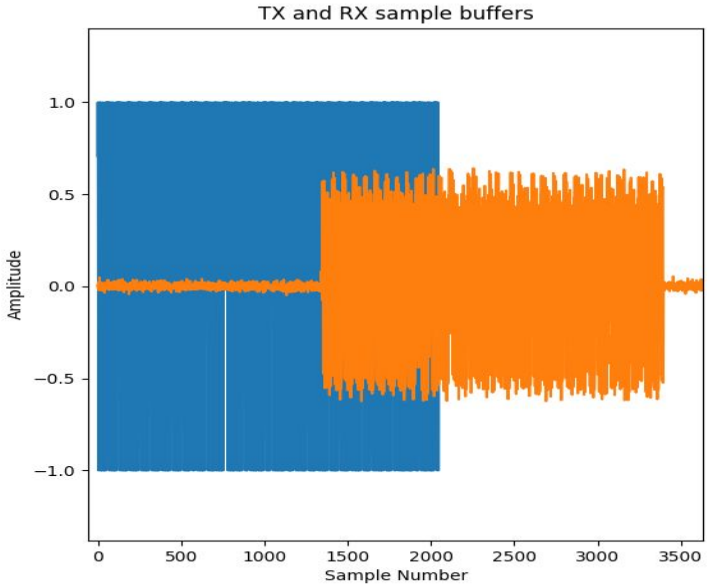

Message Source
Data
MAC
MAC Header
Data
CRC
PHY
Preamble
Mac PayLoad
PHY
GNU Radio Processing Chunks
IQ Samples Stream
Batch Size
LimeSDR Driver
Header
Data
Header
Data
FPGA Packet
FPGA Packet
USB Transaction
USB Controller
TOKEN
Data Packet
Data Packet
Hanshake Packet
Cypress EZ-FX3

Timing Measurements Data Points:

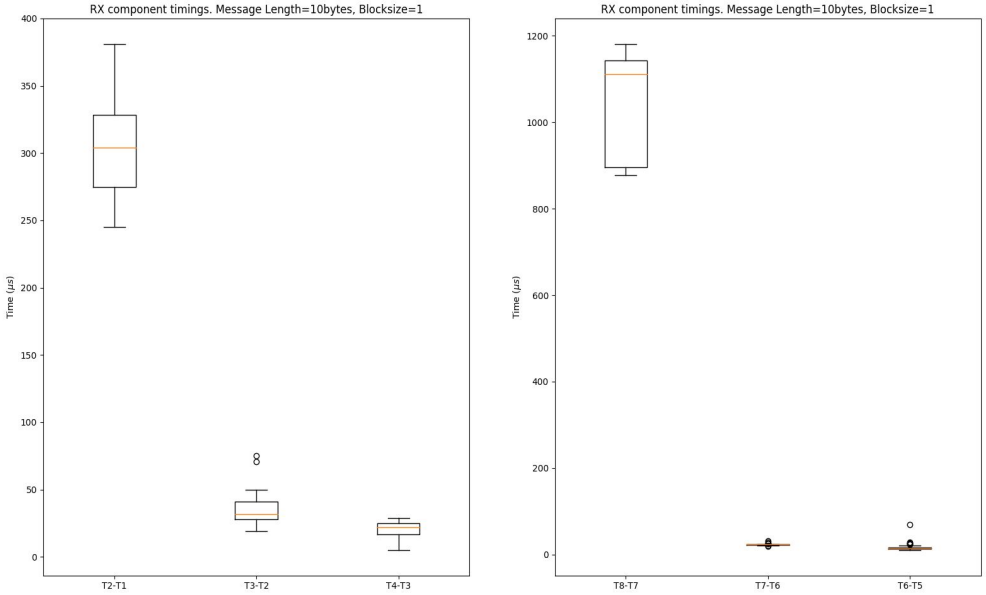


USB Host Controller Data points:

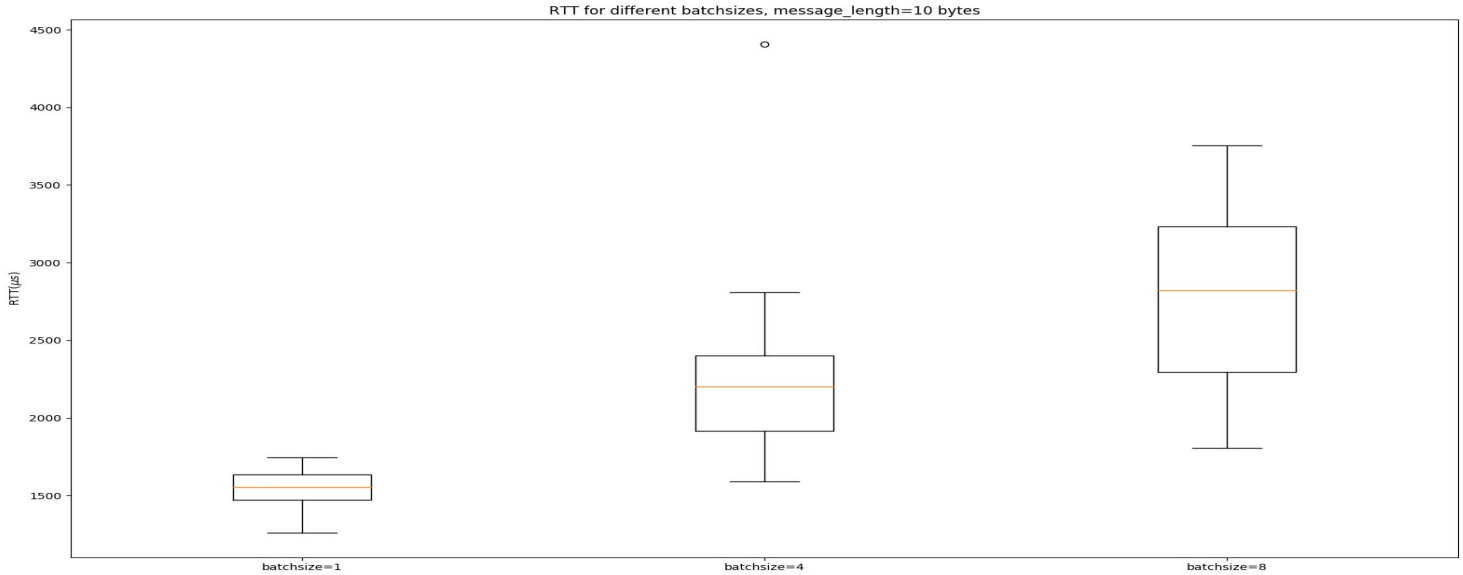




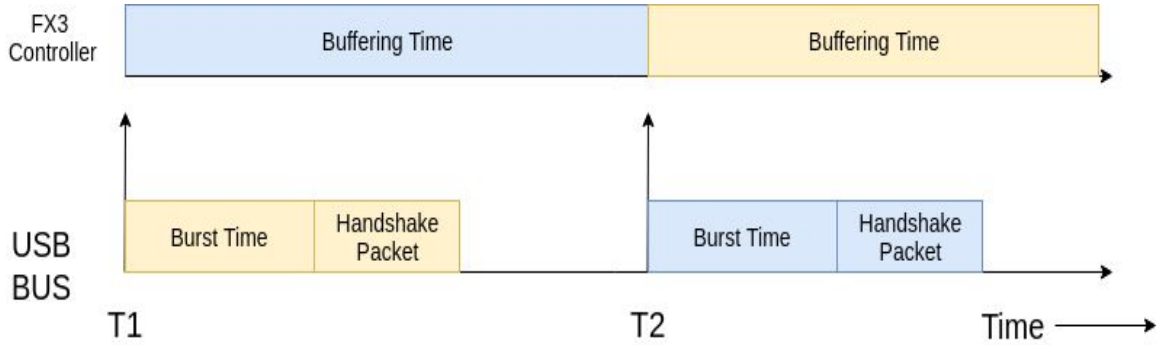
Component Timing results:



RTT vs Batchsizes:



Closer look at buffering at the USB transaction:



Buffering Time = Buffer_Length/ Sampling Rate*2* Bytes per sample
Buffer_Length= Batchsize* FPGA_Packet_Length

Burst Time = Buffer_Length / Available Bandwidth (Max: 4Gbits/sec)

Timings, batchsize=1, samp_rate= 4MSPS

Buffering time(worst case) = $4096/4*10^6*2*2$ [1 sample=2 Bytes]

$= 256\ \mu s$

Burst Time(Best case) = 4096/4 Gbits $\cong 8.192\ \mu s$

Timed Transmission

- Start transmission at particular time instant of SDR time.
- Challenge:

  SDR time □ System Time.