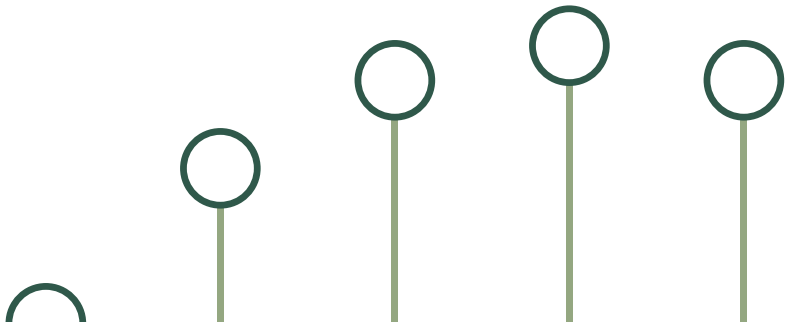
Ettus Research

Third Generation USRP Devices and the RF Network-On-Chip

Leif Johansson
Market Development
RF, Comm and SDR



About Ettus Research

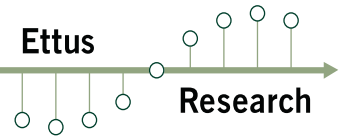
Ettus Research

- Leader in software defined radio and signals intelligence

- Maker of USRP™ (Universal Software Radio Peripheral):
  - Enables rapid development of SDR and RF systems
  - Supported by a strong software ecosystem
  - DC-6 GHz, MIMO capability, Embedded, USB/GigE
  - Wireless Innovation Forum – 2010 Technology of the Year

- About The Company
  - Founded in 2004
  - Located in Santa Clara, CA – Silicon Valley
  - Stand alone subsidiary of National Instruments since 2010

National Instruments



- *Leaders in Computer-Based Measurement and Automation*
- Long-term Track Record of Growth and Profitability
- $1.05 B Revenue in 2011
- Invest 16% Sales into R&D
- > 6,500 employees; operations in 49+ countries
- *Fortune's* 100 Best Companies to Work For 13 Consecutive Years
- Significant investments in RF test microwave design and software defined radio

VSAs & VSGs

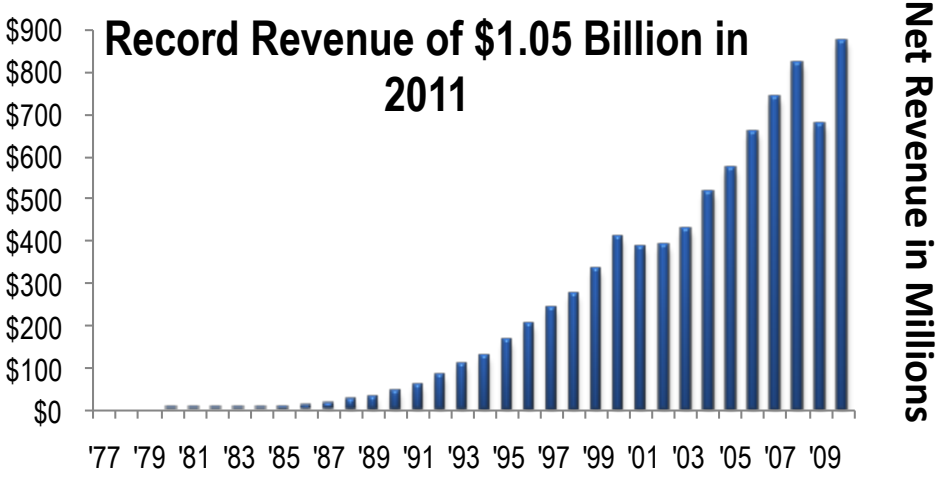
Power Meters

FPGA I/O & Co-processing

Amplifiers & Attenuators

Switching

Software Defined Radio



System Architecture

Ettus Research

Application

LabVIEW	*Custom*	**GNURadio** Python / GRC	**The Mathworks**

UHD Driver

Windows	MacOS	Linux	Embedded Linux

Hardware

Motherboard

(FPGA, host connection)

RF board

Antenna

Antenna VERT400

RF Board (WBX)

USRP N210

THIRD GENERATION USRP DEVICES

USRP FPGA Capability

	Gen1	Gen2	Gen3 E300	Gen3 X310
FPGA	CycloneI	Spartan3	Zynq	Kintex7
LogicCells	12K	53K	85K	406K
Memory	26KB	252KB	560KB	3180KB
Multipliers	NONE!	126	220	1540
ClockRate	64MHz	100MHz	200MHz	250MHz
TotalRFBW	8MHz	50MHz	128MHz	640MHz
Freespace	none	~50%	~75%	85+%

Third Generation of USRP

Easy integration of multiple processing paradigms, especially large FPGA fabric

Very large bandwidth (56 to 160 MHz), very wide frequency coverage

Massive MIMO scalability

Three family members to start

- High integration and ease of use, extremely low cost – B200/B210
- Low power, handheld MIMO – E300
- High capability and extreme performance and expandability – X300/X310

RF Network on Chip Architecture

- Common FPGA design allows for portable IP development
  UHD will expose more FPGA capabilities to the user, including partial reconfiguration
- Enable easy FPGA programming without hand written Verilog

X300 and X310

160 MHz RF Bandwidth

200 MS/s 14-bit ADCs, 800 MS/s 16-bit DACs

Dual 10GbE with SFP+ ports

- SFP+ ports will also do 1 GbE and CPRI

PCIe x4 (over cable)

Large Kintex 7 FPGA

- 840/1540 DSP units (X300/X310)

1 GB onboard DDR3

Built-in GPS Disciplined OCXO, or use with external references

2x2 MIMO w/beamforming out of the box

- Expandable to arbitrary width MIMO

Uses standard Ettus Research high dynamic range RF boards covering DC to 6 GHz

X300

Ettus Research

OctoClock & OctoClock-G

8-Way clock reference and time distribution

10 MHz and PPS to 8 USRPs

External reference or Internal GPSDO (OctoClock-G)

1U Rackmount enclosure

Ideal for Massive MIMO applications

Ettus Research

E300 Front View

E300

- Xilinx Zynq FPGA/Processor
  - Dual Core ARM Cortex A9 at 800 MHz
  - Large Programmable Logic Area
  - 1 GB DDR3 DRAM for processor, separate 512 MB for Logic
- Integrated RF
  - 70 MHz to 6 GHz integrated RF
  - 2x2 MIMO, Full Duplex
  - 56 MHz RF BW
  - Full coverage RF Filterbanks
- Gigabit Ethernet, USB 2.0 Host
- GPS Synchronization
- Stereo audio in/out
- 10-axis IMU (Accelerometers, gyros, magnetometers, altimeter)

E300 Features

Small handheld enclosure, 60mm by 120mm by 27mm

- Cellphone sized, but thicker

Battery powered

Runs a full Linux distribution maintained by key members of embedded Linux community

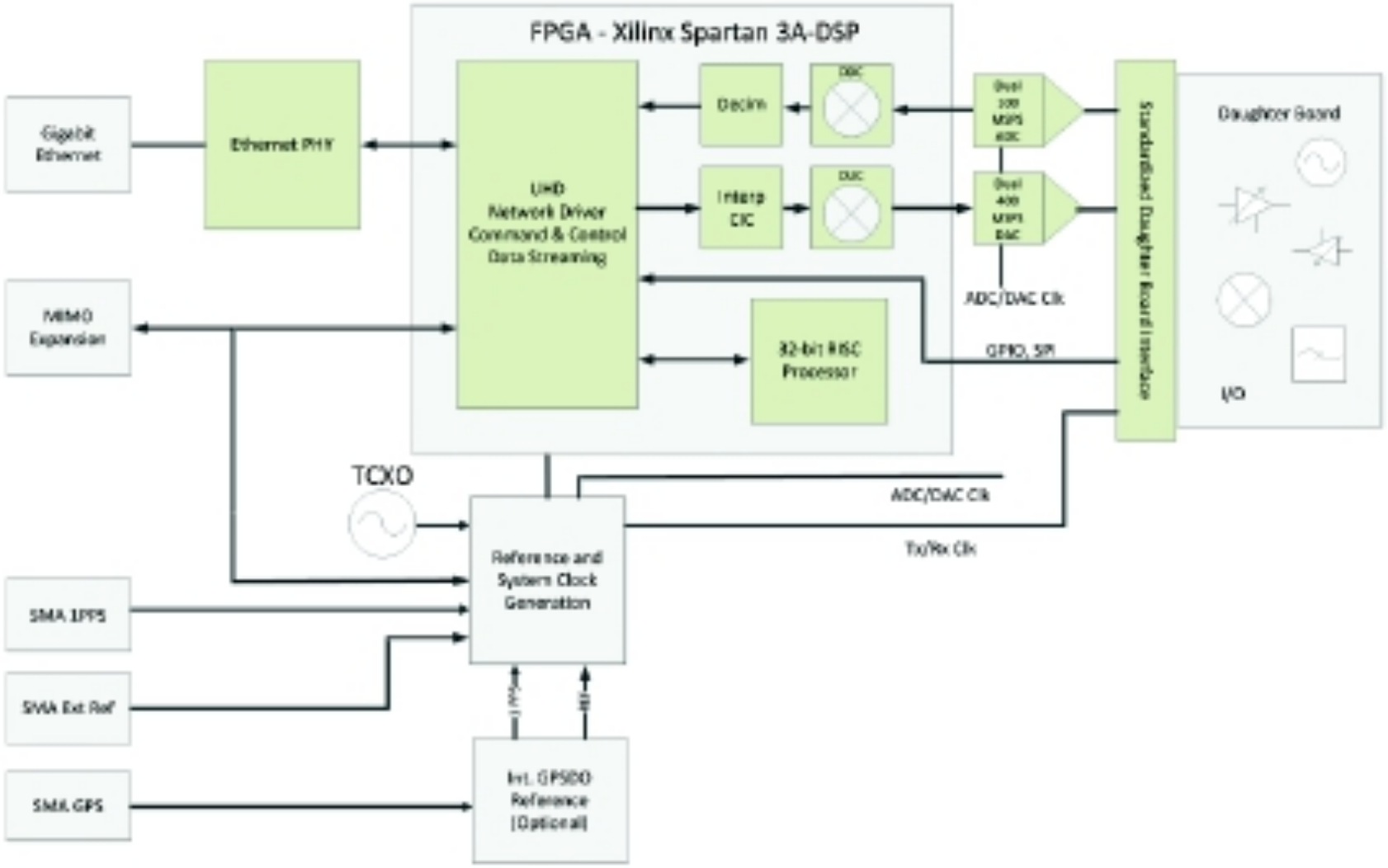
Strong SW Ecosystem

- UHD Device
- GNU Radio support
- LabVIEW and LabVIEW FPGA support
- Develop on desktop, deploy on embedded device

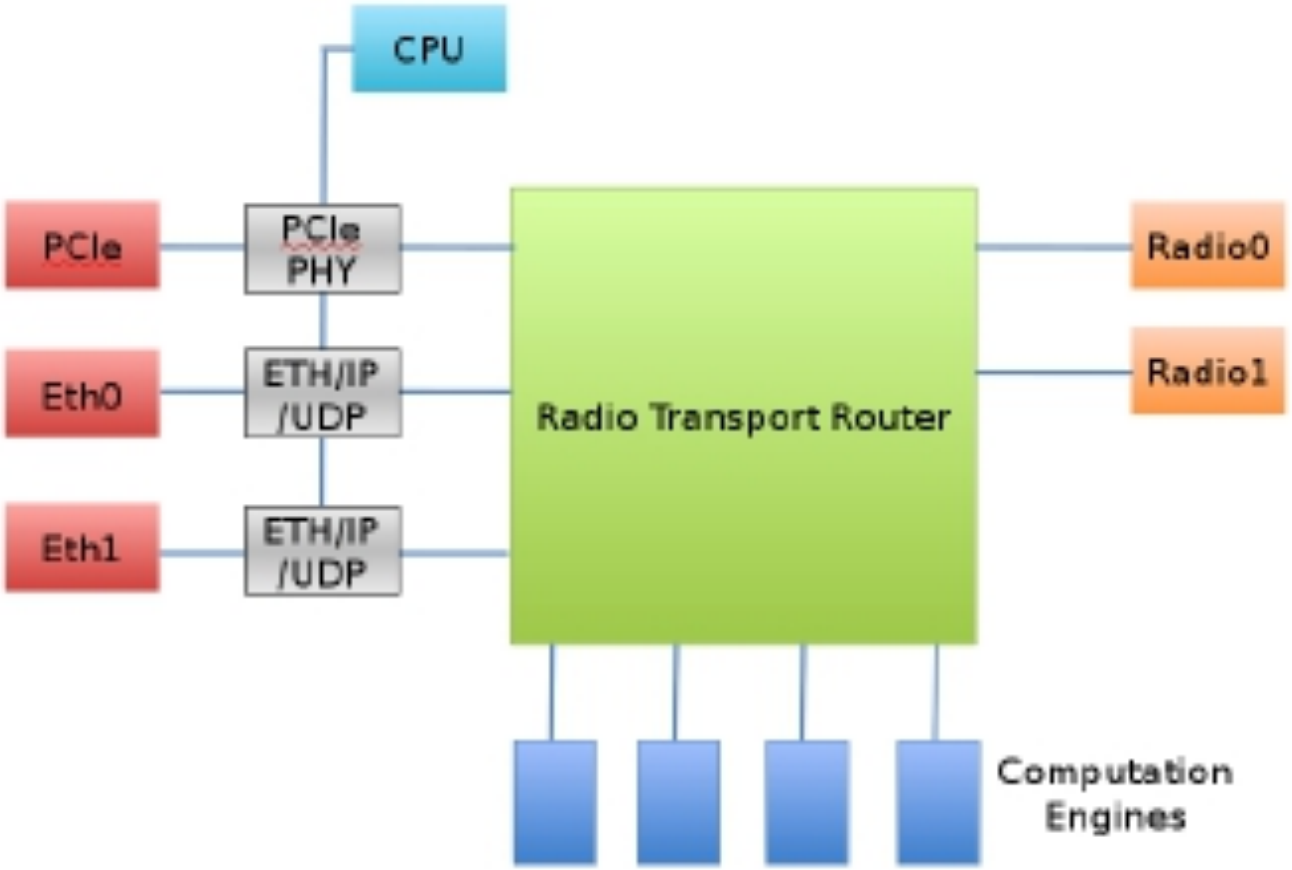
E300 PCB

RF NETWORK-ON-CHIP

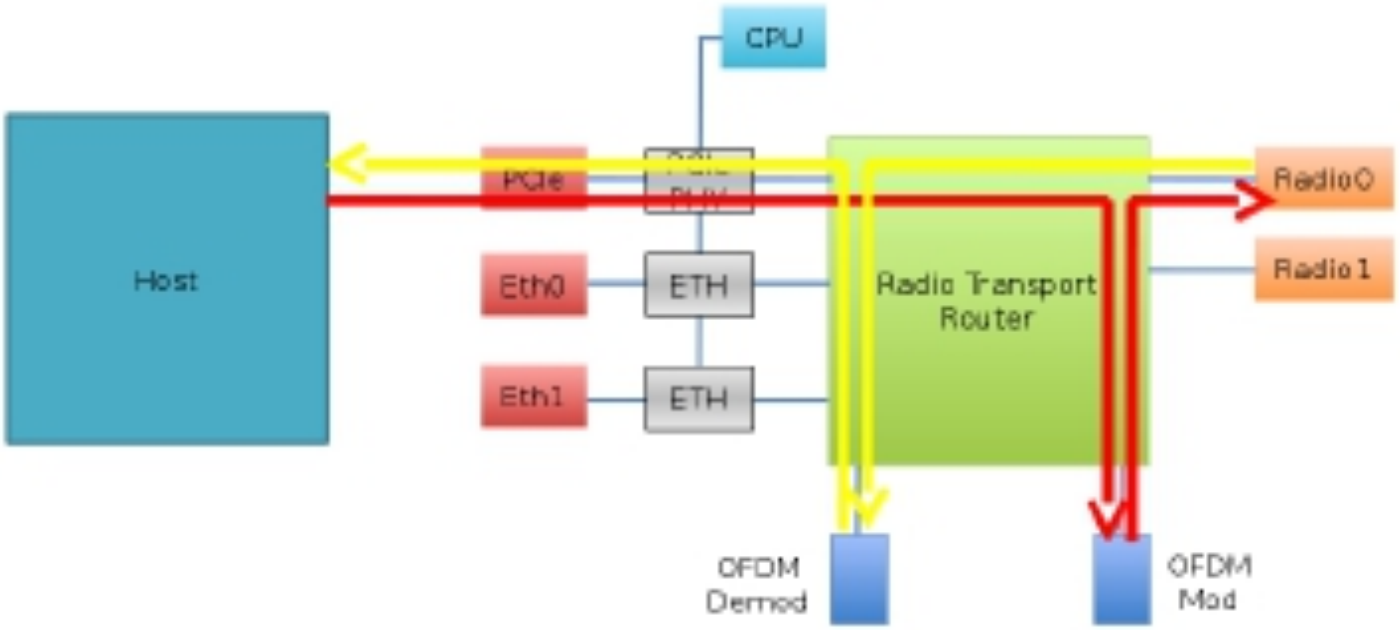
Traditional FPGA Data Flow



RF NoC



RF NoC Example Application



RF NoC Principles

Very simple Interface/API

- 64-bit AXI FIFO in and out
- AXI DDR3 interface for scratch space if necessary
  VITA-49 standard packet formats

All communication is packet-oriented over FIFO interfaces

Data (baseband samples, symbols, packets, etc.) and Control are carried over the same path

Data and Control carried in the same packet format

All endpoints are created equal

- Any-to-any communications
  No “host” is necessary

All communication is flow-controlled (both fine and coarse)

Each block can be in its own clock domain

External Interfaces

Packets entering or leaving FPGA via

- 1G/10G Ethernet
  PCIe
- USB3
- AXI interfaces to ARM and System RAM in Zynq
  Adaptation layer to other processing paradigms (i.e. massive multicore, GPU, etc.)

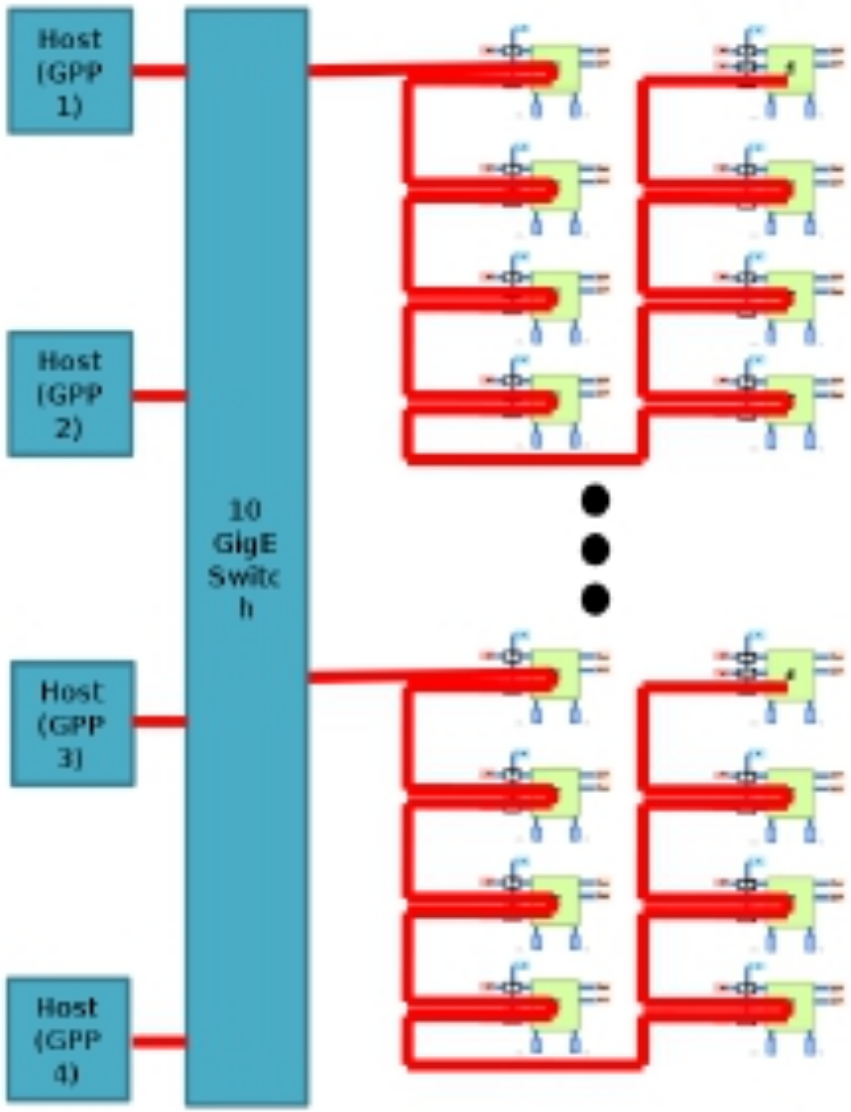
Filters out non-RFNoC traffic and passes it to the control processor

- e.g. ARP, Ping
- Out-of-band control like setting up the network router, misc. hardware controls, etc.
  Network diagnostics

Strips off other protocol layers and compresses headers on ingress, reverse operation on egress

Maps RFNoC address to external protocol address (i.e. MAC and IP addresses and UDP port)

Massively Scalable MIMO



End to End Flow Control

Uses coarse-grained (packet-oriented) flow control

- Data consumers report consumption of data packets back to producers with special flow control packets, inband
- No producer may send data to a consumer unless it knows there is enough buffer at the consumer to hold it
  - Guarantees no blocking in the routing fabric

If any block in a chain originating from a timed source (i.e. radio receiver) can't keep up, samples back up to the source

- Source reports overflow into the chain when its output buffers fill

If any block in a chain ending at a timed sink (i.e. radio transmitter) can't produce data fast enough, the pipe empties

- Destination reports underflow back up the chain towards the source when its input buffers are empty

Status

Working implementation of radios, network fabric, external interfaces, and flow control across multiple product lines with a single code base

A small number of simple computation engines have been implemented

To Do

- Implement more interesting computation engines
- Demonstrate multi-FPGA flowgraphs
- Automate the setup and routing process based on a GNU Radio flowgraph
- Automatically migrate processing from host to computation engines
- Automatically build FPGA image with user selected computation engines
  - Take advantage of partial reconfiguration
- Produce skeleton reference computation engines in various design environments
  - Verilog, VHDL, MyHDL, LabVIEW FPGA, Simulink,

Ettus Research

Thank You