RFNoC

Network on Chip for Easy FPGA Development

Matt Ettus
<matt@ettus.com>

Ettus Research

2015

USRP FPGA Capability

	Gen 1	Gen 2	Gen 3 E300	Gen 3 X310
FPGA	Cyclone I	Spartan 3	Zynq	Kintex 7
Logic Cells	12K	53K	85K	406K
Memory	26KB	252KB	560KB	3180KB
Multipliers	NONE!	126	220	1540
Clock Rate	64 MHz	100 MHz	200 MHz	250 MHz
Total RF BW	8 MHz	50 MHz	128 MHz	640 MHz
Free space	none	~50%	~75%	85+%

The Challenges

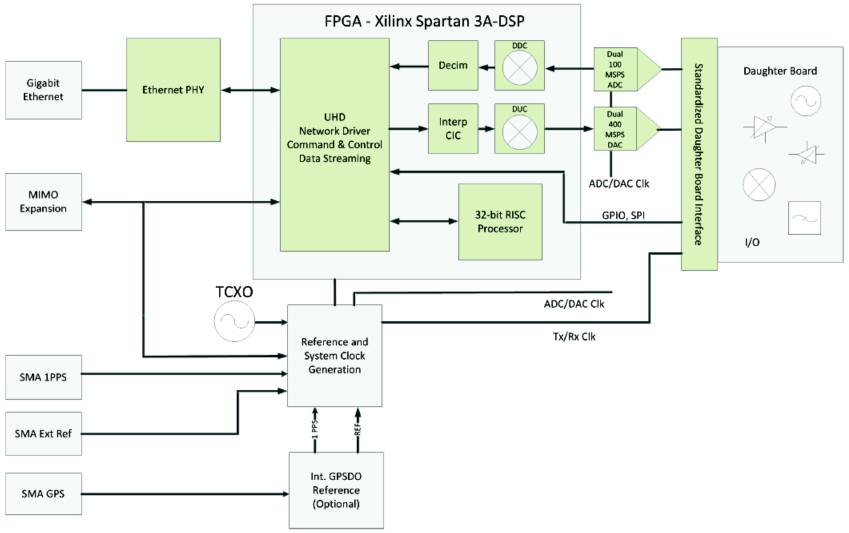
- FPGA computational capability scales with Moore’s Law but our ability to use it does not
- Domain experts in algorithms not necessarily skilled in FPGA design, networking issues, verilog, or the internals of a particular radio system
- Poor reusability of computational elements across multiple designs due in part to lack of standard interfaces
- Difficulty migrating CPU code to FPGA
- FPGA reconfigurability is on a very slow time scale
  - Seconds to load a configuration
  - Minutes to hours to compile a new configuration
  - Hours to months to develop a new design
- Difficulty scaling designs beyond one FPGA
- Highly interconnected large designs make it difficult to meet timing requirements and slow down compile cycles

The Opportunities

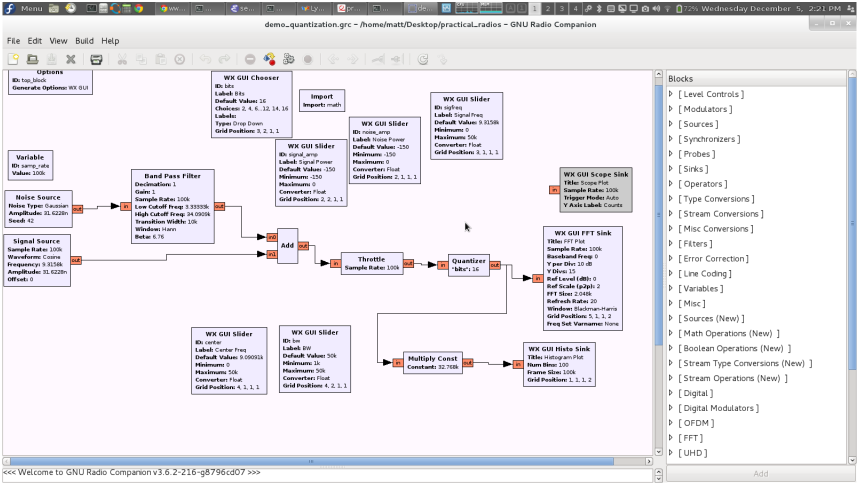
- Massive new FPGAs allow a lot of freedom
  - Don’t have to design like a “rewritable ASIC”
- Considering and valuing *composability* of blocks and code reuse early in the process can lead to large productivity gains
- Be able to take advantage of high level design languages and tools
  - can provide a better way for algorithm experts to express computation than in Verilog or VHDL
  - MyHDL, System Verilog, LabVIEW FPGA, VivadoHLS, Coregen, Simulink, etc.

Ettus Research

Traditional FPGA SDR Data Flow



The Vision



demo_quantization.grc – /home/matt/Desktop/practical_radios – GNU Radio Companion

Options
ID: top_block
Generate Options: WX GUI

WX GUI Chooser
ID: bits
Label: Bits
Default Value: 16
Choices: 2, 4, 6...12, 14, 16
Labels:
Type: Drop Down
Grid Position: 3, 2, 1, 1

Import
Import: math

WX GUI Slider
ID: sigfreq
Label: Signal Freq
Default Value: 9.3158k
Minimum: 0
Maximum: 50k
Converter: Float
Grid Position: 3, 1, 1, 1

WX GUI Slider
ID: noise_amp
Label: Noise Power
Default Value: -150
Minimum: -150
Maximum: 0
Converter: Float
Grid Position: 2, 1, 1, 1

WX GUI Slider
ID: signal_amp
Label: Signal Power
Default Value: -150
Minimum: -150
Maximum: 0
Converter: Float
Grid Position: 2, 2, 1, 1

Variable
ID: samp_rate
Value: 100k

WX GUI Scope Sink
Title: Scope Plot
Sample Rate: 100k
Trigger Mode: Auto
Y Axis Label: Counts

Noise Source
Noise Type: Gaussian
Amplitude: 31.6228n
Seed: 42

Band Pass Filter
Decimation: 1
Gain: 1
Sample Rate: 100k
Low Cutoff Freq: 3.33333k
High Cutoff Freq: 34.0909k
Transition Width: 10k
Window: Hann
Beta: 6.76

Signal Source
Sample Rate: 100k
Waveform: Cosine
Frequency: 9.3158k
Amplitude: 31.6228n
Offset: 0

Add

Throttle
Sample Rate: 100k

Quantizer
"bits": 16

WX GUI FFT Sink
Title: FFT Plot
Sample Rate: 100k
Baseband Freq: 0
Y per Div: 10 dB
Y Divs: 15
Ref Level (dB): 0
Ref Scale (p2p): 2
FFT Size: 2.048k
Refresh Rate: 20
Window: Blackman-Harris
Grid Position: 5, 1, 1, 2
Freq Set Varname: None

WX GUI Slider
ID: center
Label: Center Freq
Default Value: 9.09091k
Minimum: 0
Maximum: 50k
Converter: Float
Grid Position: 4, 1, 1, 1

WX GUI Slider
ID: bw
Label: BW
Default Value: 50k
Minimum: 1k
Maximum: 50k
Converter: Float
Grid Position: 4, 2, 1, 1

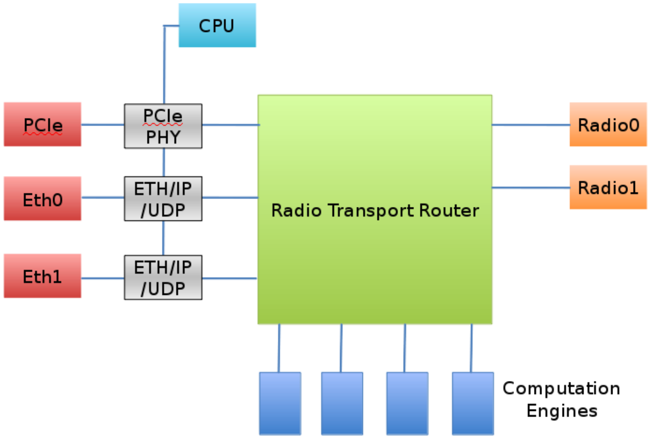
Multiply Const
Constant: 32.768k

WX GUI Histo Sink
Title: Histogram Plot
Num Bins: 100
Frame Size: 100k
Grid Position: 1, 1, 1, 2

```
<<< Welcome to GNU Radio Companion v3.6.2-216-g8796cd07 >>>
```

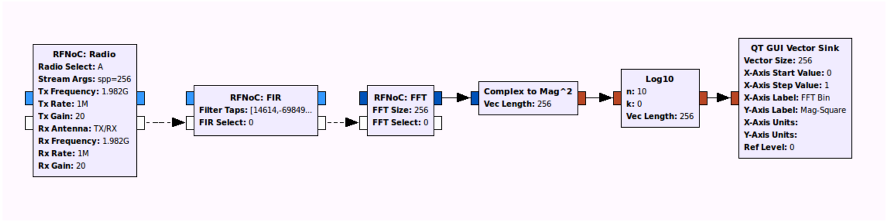
Ettus Research

RF NoC



Ettus Research

RFNoC in GNU Radio



RFNoC: Radio
Radio Select: A
Stream Args: spp=256
Tx Frequency: 1.982G
Tx Rate: 1M
Tx Gain: 20
Rx Antenna: TX/RX
Rx Frequency: 1.982G
Rx Rate: 1M
Rx Gain: 20

RFNoC: FIR
Filter Taps: [14614,-69849...
FIR Select: 0

RFNoC: FFT
FFT Size: 256
FFT Select: 0

Complex to Mag^2
Vec Length: 256

Log10
n: 10
k: 0
Vec Length: 256

QT GUI Vector Sink
Vector Size: 256
X-Axis Start Value: 0
X-Axis Step Value: 1
X-Axis Label: FFT Bin
Y-Axis Label: Mag-Square
X-Axis Units:
Y-Axis Units:
Ref Level: 0

RF NoC Principles

- Distributed asynchronous implementation of Kahn Process Networks
- Protocol has some similarities to Rapid IO
- Very simple Interface/API – 64-bit AXI FIFO in and out
- 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

Radio Blocks

- Contain everything which must happen in the sample clock domain
  - All precise timekeeping and sequencing
  - All control of radio hardware
  - All sample-rate DSP (DUC, DDC, IQ Balance, most sample rate conversion)
- Produces and consumes packets of baseband samples at a fixed rate

Computation Engines

- Can perform arbitrary processing on both data and control
  - Generic FIR engine
  - FFT machine
  - OFDM sync
  - Turbo decoding
  - Frequency hopping state machine
  - AGC state machine
  - Protocol processing
  - Upper layer stacks
  - CPU (i.e. Microblaze)
  - Large memory buffers
  - etc.

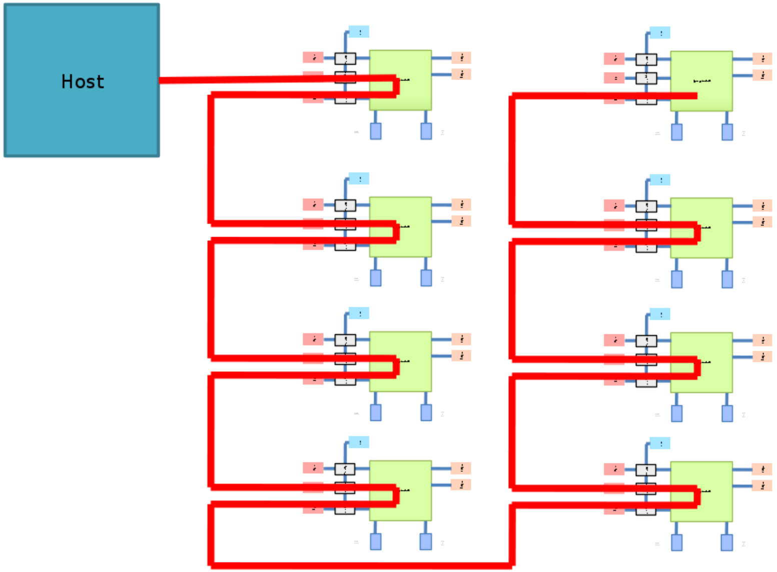
External Interfaces

- 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)

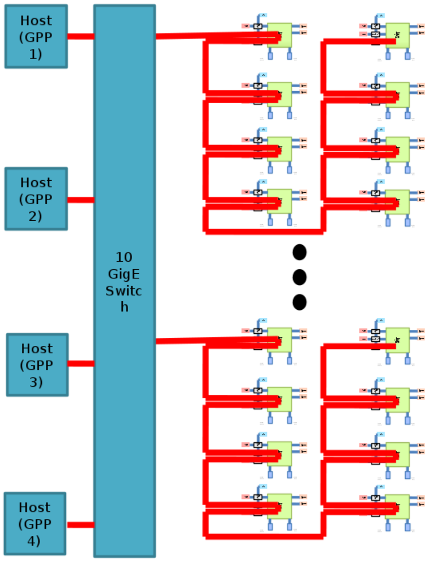
Network Fabric

- Full crossbar to route packets between blocks
- Routing tables are loaded out of band by the control processor, and are set up by the application
- All routing decisions are made entirely based on the first line of every packet
- No packet is allowed into the crossbar until it is complete
  - Prevents slow packets from causing congestion
  - Ensures data flows through at full rate
- Flow control assures that any packet entering the router can also exit because there is buffer space guaranteed on the other side

Simple MIMO Example



Massively Scalable MIMO



Status

- Working implementation of radios, network fabric, external interfaces, and flow control across multiple products with a single code base
- A number of interesting computation engines have been implemented
- Automatic setup and routing based on a GNU Radio flowgraph
- Automatically builds FPGA image with user selected computation engines
- To Do
  - Implement more interesting computation engines
  - Demonstrate multi-FPGA flowgraphs
  - Take advantage of partial reconfiguration
  - Produce skeleton reference computation engines for other design environments
    - MyHDL, LabVIEW FPGA, Simulink, VivadoHLS, etc.
  - Automatically migrate processing from host to computation engines