CASPER / PIRE DSP Summer School
Lecture Outlines 12-13 Aug. 2019

v. 19aug10

D. Marone

Introduction to Radio Astronomy

+ Characteristics of Radio Astronomy
  - Frequency/Wavelength ranges
    Atmospheric and ionospheric bounds
  - Special characteristics
    Long wavelengths, diffraction, single-mode
    Coherent detection, field sensing
  - Early radio astronomy history
  - Radiation processes and science motivations
  - Radio telescope morphologies
    Low/high frequency
    Single-aperture and interferometry
    Phased arrays
  - Detection Methods
    Coherent detection
    Mixing and Amplification
    Incoherent detection

+ Radiation and Detection
  - Brightness, Flux density
  - Blackbody, RJ limit, and brightness temperature
  - Thermal noise in circuits, equivalent temperature
  - Square-law detection of voltage
  - Radiometer equation

Introduction to Interferometry

+ Telescopes
  - Illumination and primary beam

+ Why Interferometry

+ Two-element Analysis
  - Response of multiplying interferometer
  - Effect of bandwidth

+ Interferometric Measurement
  - Fourier Transform response (van Cittert-Zernike)
  - FT relationships in interferometric imaging

D. Gorthi

Fourier Transforms I:

1. Fourier Series
2. Fourier Transforms
    2.1 Common Fourier Pairs
    2.2 Properties of transforms
3. 2D Fourier Transforms

Fourier Transforms II:

1. Sampling
2. Discrete Time Fourier Transform
3. Discrete Fourier Transform
4. Fast Fourier Transform

J. Moran

D. Werthimer

Radio Astronomy Signal Processing Instrumentation and CASPER

+ Why astronomers need the following instruments, and what do these instruments measure:

-spectrometers
-pulsar/frb/seti spectrometers
-beamformers
-correlators

+ Crude block diagrams and ways to implement the above instruments

+ Intro to casper: (architectures, ethernet, simulink libraries, open source, tutorials,
casper boards, commercial boards, and example instruments)

J. Hickish

K. Haworth

FPGA Basics

What is an FPGA?

- "Gates", "programmable" definition
- Concept of bitfile
- Comparison to CPU, GPU, ASIC
- Why choose an FPGA
- Interface design, state machines, pipelines

FPGA Design Flow

- Requirements
- Architecture/Specification
- Code (with explanation of hardware vs software language, IP blocks)
- Simulate
- Synthesis
- Place and Route
- Bitfile
- Lab Test

V. van Tonder

Introduction to GNU/Linux

* What is GNU/Linux
* What is a Shell
  * stdin, stdout, and stderr
  * echo
  * ls
  * Tab complete
* File System
  * Tree like file system
  * cd, pwd, chmod, touch, executables
  * cat, mkdir, rm, cp, mv, diff, cp
  * wildcards
* IO Redirection & Searching
  * > , < , >> , |
  * grep , cut
* Miscellaneous
  * ln , ps , kill, apt-get, sudo, su , ssh, scp
  * Shortcuts ctrl+r, ctrl+a, ctrl+e, ctrl+c, ctrl+d

References

1. INTRO(1) Introduction to user commands manual page

1. Open your bash terminal using ctrl+alt+t
2. At the prompt $ type man intro

1. What is GNU/Linux, "Debian GNU/Linux Installation Guide"
2. Chapter 2 of "Unix Text Processing" by D Dougherty & T O' Reilly
3. Navigation, "Linux command line" by W Shotts
4. Bash Basics, "Learning the bash Shell" by C Newham & B Rosenblatt
5. https://www.gnu.org/
6. https://www.kernel.org/

TcpBorphServer

This talk will consist of a couple of slides, followed by a demonstration.

Slides (~8 min):
What is KATCP?
What is TcpBorphServer?
How does the TcpBorphServer software fit into the picture
Programming the FPGA
Fpg file: explain the meta data
Kcpfpg utility
Uploadbin
Control and Monitoring
listdev
Read/write from memory
kcpcmd utility

Demonstration (~12 min):
This talk happens directly after the "Introduction to Red Pitaya and support in the CASPER toolflow".
Therefore a RP will be connected and available. This will be setup the night before.
Open a .fpg file and look at meta data. Explain the terminology.
Program a RP with a fpg file that's located on the ARM. ie another RP, not the one that has been used in the previous demo.
Demonstrate reading and writing using kcpcmd.

A. Martens

DSP with FPGAs

1. DSP with an FPGA basics. (vs traditional processors, GPUs etc)
   - high configurability (resource management)
   - variable clock speed (placement, timing)
   - debugging not built in
2. Why do DSP with an FPGA? (as opposed to something else)
   - low latency
   - high IO vs compute ratio
   - configurability
   - efficiency
3. How to do DSP with an FPGA? (CASPER)
   - parallelism

- abstraction versus efficiency
  - graphical data flow representation vs HDL
  - standard hardware (BEE, iBOB, ROACH, SKARAB, SNAP ...)
  - standard libraries and interfaces

4. Quantization
   - configurable bit widths (rounding strategies)
   - dynamic range
   - linearity
   - floating point
5. Scaling DSP
   - push to more antennas, more bandwidth, more channels
   - networking
   - timing and synchronization

J. Dowell

After the FPGA: Building DSP Processing Pipelines on a Non-real-time Platform

+ Hybrid FPGA/GPU Architectures
- What are they?
- Advantages over other approches
- Example: LWA Sevilleta

+ Bifrost
- What is Bifrost?
- Key concepts
  - * ring buffers
  - * sequences/spans/gulps
  - * blocks
- What can it do
- Examples: LWA-SV pipelines and EPIC

+ How to build a hybrid system with Bifrost
- Networking
- Servers
- GPUs
- Software
- IPC

+ Tuning, testing, and pitfalls
- memory and core binding
- IRQ binding
- gulp size and data type
- division of labor
- unit tests
- canned data (offline testing)
- Bps vs. IOPS

+ Wrapup
- Where Bifrost is going
- Where LWA is going with Bifrost

A. Renard

Building a high performance DSP framework for GPUs and x86 hardware: A look at the design of the CHIME X-engine and beamformer.

* Examples of hybrid DPS systems
  * Overview of the CHIME correlator
    * System hardware
    * Hybrid corner turn
    * Correlation
    * Beamforming
  * Capture systems

* Introducing Kotekan
  * Core system design elements
    * stages
    * buffers
    * data views
    * command objects
  * Configuration design
  * HTTP Server
  * Prometheus integration

* Example configurations and stages
  * Packet capture config
  * Extending kotekan
    * Adding new stages
    * Writing new configs
    * Adding GPU operations
  * Built-in buffer operations

* Performance optimization
  * DPDK for packet capture
    * Userspace drivers
    * Cache aware memory operations
  * Thread/Memory management
  * GPU Pipelining
  * Low bit-depth GPU ops
  * AVX for CPU operations
  * Threading vs Event-driven programming
  * Common issues with software DSPs

* Looking forward
  * What is coming to kotekan
  * Next generation systems

J. Manley

Packetized Instruments and Networking

This will be a crash-course into basic Ethernet technologies, including OSI layers and cables. I'll touch briefly on some of protocols used in radio astronomy systems, and the challenges faced when scaling-up ethernet systems.

* Radio astronomy interconnect challenges
* Why ethernet?
* Routing types (unicast, broadcast, multicast, anycast)
* OSI model/layers
* Ethernet versions/speeds
* Cabling: copper, breakout/octopus, AOC, MMF, SMF, CWDM vs DWDM, silicon photonics
* L2 vs L3 operation
* Subnets and gateways
* IPv4 vs IPv6
* UDP vs TCP
* Highlight some common protocols: ARP, DHCP, DNS, ECMP, IGMP, OSPF
* Multicasting at L2 and L3
* Debugging Ethernet exchanges
* Scaling Ethernet: common industry practices; unique RA requirements
* Clos networks
* Drawbacks of big chassis
* Buffering and collisions
* Ethernet on CASPER FPGAs
* Future: SDN opportunities?

W. New

Red Pitaya and the CASPER Toolflow

Why the Red Pitaya?

1. Casper has a range of FPGA based boards but they are all expensive. This is a large barrier to entry.
2. RP is cheap, has ADCs and DACs built in.
3. Zynq architecture
4. Great platform to get anyone who is interested, working with the CASPER tools.
5. Red Pitaya already has good software support for instruments such as an oscilloscope and a signal generator. Which extends it uses.

1. What is the Red Pitaya
   Image of the Red Pitaya
   a. Zynq
   Diagram of PS and PL
   b. FPGA
   c. ADC/DACs
   d. SOC architecture
2. CASPER Ecosystem for FPGA design
   a. Mlib_devel

    b. Fpg files
    c. CASPERFPGA
    d. TCPBorphServer
    e. KATCP
3. Support for the RP
    a. Software Registers
    b. Programming
    c. BRAMs
    d. ADCs/DACs
    e. Clocking and Resets
        MMCMs to create the requested frequencies.
    f. AXI Interface
        Explain protocol
            Address based
        Automatic memory address generation
        Leveraging Matts/Riccardos code

T. van Balla

A gentle introduction to casperfpga

This tutorial presents the basic building blocks (methods) needed to control new hardware via casperfpga. This tutorial is delivered as a hybrid between a lecture and hands-on walkthrough/demonstration.

● Introduction to casperfpga
● A high-level overview of how casperfpga works
    ○ Server-side
    ○ Client-side
● Transport Protocols
    ○ skarab_transport
    ○ katcp_transport
    ○ Supporting different transport protocols ('Automagically')
● Building a transport protocol
    ○ Implementation on hardware (brief overview)
■ skarab_transport - MicroBlaze
■ katcp_transport - TcpBorphServer
    ○ Implementation in casperfpga (in depth)
■ Basic structure
■ Reading/Writing Registers
■ Programming casper hardware
■ Platform unique functions/features
■ Tips, Tricks, and Troubles
■ Testing. Testing. Testing.
● Wrapping up