On the Integration of OpenCL into a Software Defined Radio

Michael L Dickens and J Nicholas Laneman
mdickens@nd.edu        jnl@nd.edu
Software Defined Radio

- Real-time signal processing, using a class of radios that can be reprogrammed and reconfigured primarily via software
- Digitize as close to the antenna as possible
- Still significantly dependent on hardware
SDR Runtime

Waveform Graph

SDR Runtime

Processing Devices (CPU, GPU, DSP)

Mapping
Buffering
Scheduling
Processing
SDR Layers Model

1. Hardware
2. Operating System
3. Runtime / Block
4. Block Processing
5. Script
6. Graphical
7. Application

Application
SQuid
Saline
Spf / naCL
Shark / naCL
Salt / naCL
Operating System
Hardware
Surfer Overview

Salt: *Surfer* Operating System and Hardware Abstraction Layer
naCL: OpenCL Abstraction Layer

<table>
<thead>
<tr>
<th></th>
<th>Operating System API, ABI, and Kernel</th>
<th>Host Hardware Abstraction Layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardware</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
1. Signal processing *flavors*, providing a new form of dynamic runtime reconfiguration and heterogeneous processing

2. Use of thresholds for scheduling, to allow direct control over throughput and latency

3. Use of runtime statistics for modifying system runtime behavior, using a Supervisor
Demodulate narrowband FM signal and downsample to typical audio rate.
1. Signal processing flavors, providing a new form of dynamic runtime reconfiguration.

<table>
<thead>
<tr>
<th>Block</th>
<th>Processing Flavors</th>
</tr>
</thead>
<tbody>
<tr>
<td>State</td>
<td>Intel CPU</td>
</tr>
<tr>
<td>Glue Table</td>
<td>Optimized</td>
</tr>
<tr>
<td>Flavor</td>
<td>GPU</td>
</tr>
<tr>
<td></td>
<td>Variant 1</td>
</tr>
<tr>
<td></td>
<td>Optimized</td>
</tr>
<tr>
<td></td>
<td>PPC CPU</td>
</tr>
<tr>
<td></td>
<td>Optimized</td>
</tr>
<tr>
<td></td>
<td>ARM CPU</td>
</tr>
<tr>
<td></td>
<td>Optimized</td>
</tr>
</tbody>
</table>
Use of thresholds for scheduling, to allow explicit control over throughput and latency.
3 Use of runtime statistics for modifying system runtime behavior, using a Supervisor

- Load
  - Host Processor
  - Surfer
  - Any Surfer Thread

- User Defined
  - Energy Use

- Block Timing
  - Queued
  - Dequeued
  - Processing Start
  - Processing End
  - Finished
Data Flow by Thread

Thread

Legend

- Preprocessing
- Processing
- Postprocessing
- Sleeping
- Notification
- Queued

GPU

CPU

Relative Time (ms)

GPUTC'12'US   2012-May-17
Data Flow by Thread

Thread

Source to Sink Latency

Legend

Preprocessing
Processing
Postprocessing
Sleeping
Notification
Queued

Relative Time (ms)
Data Flow by Block

Legend

Preprocessing
Processing
Postprocessing
Waiting
Notification

Thread

Legend

Preprocessing
Processing
Postprocessing
Waiting
Notification

Relative Time (ms)

0 0.05 0.1 0.15 0.2 0.25 0.3 0.35 0.4

Sink
DS
QD
Source

Thread

0 1 0

0 1

Source

Thread

0 1 0

0 1

Source
Data Flow by Block

Thread

Source to Sink Latency

Legend

- Preprocessing
- Processing
- Postprocessing
- Waiting
- Notification

Source

0

Sink

0.4

0.35

0.3

0.25

0.2

0.15

0.1

0.05

0.0

Relative Time (ms)

0

0.05

0.1

0.15

0.2

0.25

0.3

0.35

0.4

0

1

DS

QD

Source

1

Throughput 3 MS/s

Latency 331 µs
OFDM Demo

- Extended audio loopback
- Keyboard-based Supervisor runtime system control
  - Rotate through flavors
  - Increase and decrease processing threshold
OFDM Demo Rx

Options
ID: demo_uhd_ofdm_demod_audio
Title: Demo doi...od -> Audio
Author: Michael Dickens
Generate Options: QT GUI

Variable
ID: samp_rate
Value: 1M

QT GUI Range
ID: audio_gain
Default Value: 1
Start: 0
Stop: 10
Step: 100m

UHD: USRP Source
Device Addr: serial=46a79a30
Mb0: Subdev Spec: A:0
Samp Rate (Sps): 1M
Ch0: Center Freq (Hz): 5G
Ch0: Gain (dB): 30
Ch0: Antenna: J1

Audio Sink
Sample Rate: 8.681k

OFDM Demod
Modulation: BPSK
FFT Length: 512
Occupied Tones: 200
Cyclic Prefix Length: 128
SNR: 10

Multiply Const
Constant: 1

QT GUI Sink
Name: QT GUI Plot
FFT Size: 1.024k
Center Frequency (Hz): 0
Bandwidth (Hz): 1M
Source to Sink Latency

Legend

Preprocessing
Processing
Postprocessing
Waiting
Notification

Instantaneous Throughput: 8.8 MS/s
Latency: 1.16 ms

Gain
FFT Shift
iFFT
GR Preamble
GR Mapper
S Packet
GR Packet
Source

Sink
CP
GR Packet Encoder

Source to Sink Latency

Legend

- Preprocessing
- Processing
- Postprocessing
- Waiting
- Notification

Source
GR Packet
S Packet
GR Mapper
GR Preamble
FFT Shift
iFFT
Gain
CP
Sink

Relative Time (ms)
0 20 40 60 80 100 120 140

Instantaneous Throughput: 1.4 MS/s
Latency: 109 ms
Conclusions

- Evolving software defined radio runtime to exploit heterogeneous processing
- Key Features: Flavors, Thresholds, Statistics and Supervisor
- Supervisor can use statistics to set thresholds and flavors, providing runtime system control of throughput / latency, and mapping
Future Work

- GUI using Qt (SQuid)
- Flavors for CUDA and other processors
- Better optimized GPU flavors
- Interval / cluster mapping
- QoS-based adaptive runtime
- Using more embedded graph information
Thank you!

Questions?