GPU Based Polyphase Filter Banks for VLBI

Mark McCurry
Christopher Beaudoin & Geoff Crew

MIT Haystack REU Program 2011
Outline

Introduction

What is being done?
PFB
Hardware

Implementation

CUDA

Results

Quality
Quantity

Future Work

Conclusions

GPU Based Polyphase Filter Banks for VLBI
What is being done?

DSP

- Input of Analog values from data source near 4 Gb/s
### What is being done?

**DSP**

- Input of Analog values from data source near 4 Gb/s
- Perform Polyphase Filter Bank (PFB) on a Nvidia GPU
What is being done?

**DSP**

- Input of Analog values from data source near 4 Gb/s
- Perform Polyphase Filter Bank (PFB) on a Nvidia GPU
- Output of channelized frequency spectrum
What is being done?

DSP

- Input of Analog values from data source near 4 Gb/s
- Perform Polyphase Filter Bank (PFB) on a Nvidia GPU
- Output of channelized frequency spectrum
- All of this done in realtime
Polyphase Filter Bank

Two Channel Polyphase Filter Bank

\[ x[n] \]
\[ \downarrow 2 \]
\[ z^{-1} \]
\[ \downarrow 2 \]

\[ E_0(z) \]
\[ \downarrow \]
\[ E_1(z) \]
\[ \downarrow \]

\[ f_1[n] \]
\[ f_2[n] \]

---

GPU Based Polyphase Filter Banks for VLBI
Roach Digital Backend (RDBE)
Nvidia GPUs
CUDA

What is CUDA?
CUDA

```c
__global__ void cu_unquantize(float *dest, const int8_t *src, size_t N)
{
    const size_t i = blockDim.x*blockIdx.x + threadIdx.x;
    if(i<N)
        dest[i] = static_cast<float>(src[i]);
}
```
// Setup via cudaMemcpy

// input and output are on the GPU
cu_unquantize<<<10,10>>>(output, input, 100);

// Return via cudaMemcpy
Quality of output: Actual

Magnitude response of 256 tap FIR, 32 tap FFT based PFB

Frequency (half-cycles/second) vs Normalized Gain

GPU Based Polyphase Filter Banks for VLBI
Quality of output: Matlab

Matlab Prototype PFB

Frequency (half-cycles/second) vs. Normalized Gain
**Speed**

Table: Performance of GPU Code

<table>
<thead>
<tr>
<th>Performance Metrics</th>
<th>Data Input Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference Implementation</td>
<td>744 MB/s</td>
</tr>
<tr>
<td>No extra channels</td>
<td>540 MB/s</td>
</tr>
<tr>
<td>Hardcoding FIR size</td>
<td>756 MB/s</td>
</tr>
<tr>
<td>Hardcoded FIR, Hand Tuned Block Size</td>
<td>890 MB/s</td>
</tr>
<tr>
<td>Using 1/10 cost 470GTX</td>
<td>637 MB/s</td>
</tr>
</tbody>
</table>

As of August 1st
Increasing Speeds

How to proceed:

- Profiling, and extensive testing
Increasing Speeds

How to proceed:

- Profiling, and extensive testing
- Effective use of CUDA resources
Increasing Speeds

How to proceed:
- Profiling, and extensive testing
- Effective use of CUDA resources
- Waiting for new cards
Conclusions

- CUDA makes heavyweight software processing possible
Conclusions

- CUDA makes heavyweight software processing possible
- Real time processing is possible at lower rates
Conclusions

- CUDA makes heavyweight software processing possible
- Real time processing is possible at lower rates
- GPU processing is applicable to Astronomy and scientific computation