

# 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



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



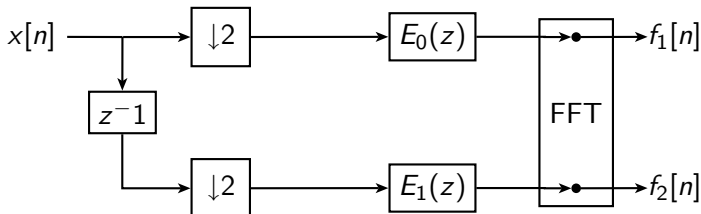
What is being done?

# DSP

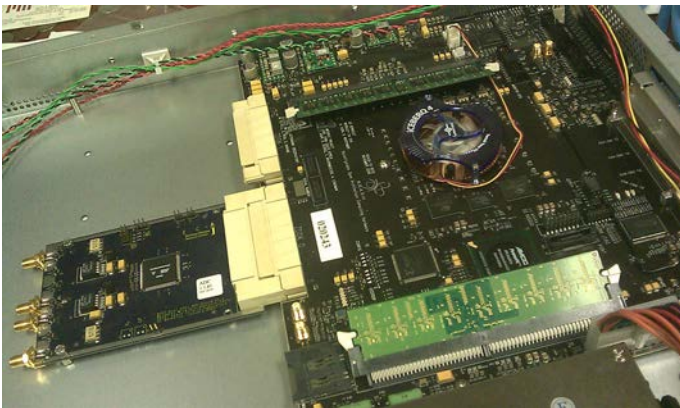
- ▶ Input of Analog values from data source near 4 Gb/s
- ▶ Perform Polyphase Filter Bank (PFB) on a Nvidia GPU
- ▶ Output of channelized frequency spectrum
- ▶ All of this done in realtime

# Polyphase Filter Bank

## Two Channel Polyphase Filter Bank



# Roach Digital Backend (RDBE)



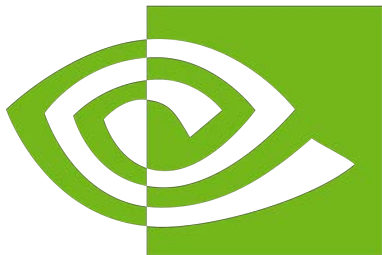




# Nvidia GPUs



# What is CUDA?



# NVIDIA®



# CUDA

---

```
__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]);
}
```

---



## CUDA 2

---

```
//Setup via cudaMemcpy
```

```
//input and output are on the GPU
```

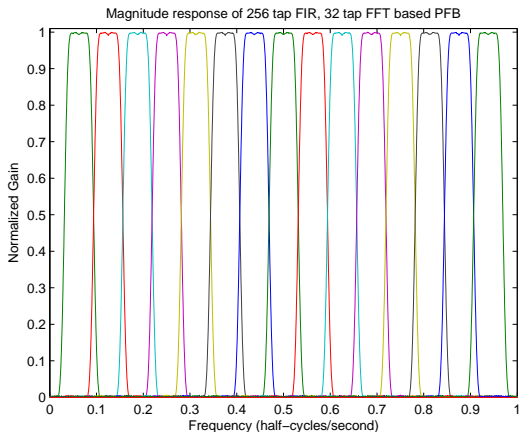
```
cu_unquantize<<<10,10>>>(output, input, 100);
```

```
//Return via cudaMemcpy
```

---

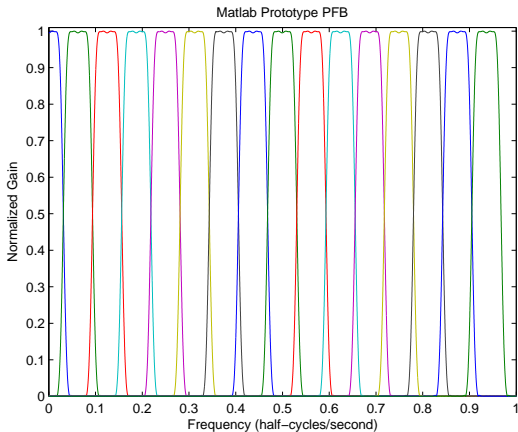


# Quality of output: Actual





# Quality of output: Matlab



# Speed

Table: Performance of GPU Code<sup>1</sup>

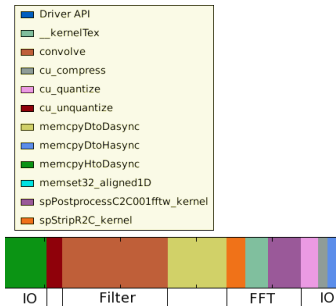
Performance Metrics	Data Input Rate
Reference Implementaiton	744 MB/s
No extra channels	540 MB/s
Hardcoding FIR size	756 MB/s
Hardcoded FIR, Hand Tuned Block Size	890 MB/s
Using 1/10 cost 470GTX	637 MB/s

<sup>1</sup>As of August 1st

# Increasing Speeds

How to proceed:

- ▶ Profiling, and extensive testing

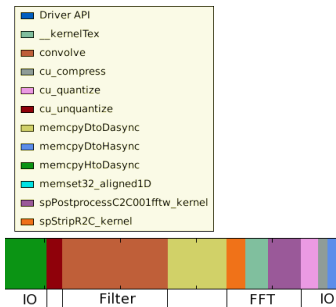




# Increasing Speeds

How to proceed:

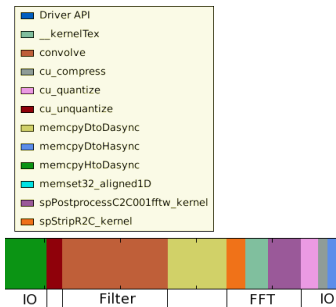
- ▶ Profiling, and extensive testing
- ▶ Effective use of CUDA resources



# 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