## The software spectrometer SpectralGpu

G. Comoretto, C. Baffa, E. Giani

The SpectralGpu project is aimed at the development of a software spectrometer for the Radio wawelengths range using the computation power of the recently available GPU video boards.

## 1 Functional Design

The SpectralGpu spectrometer gets its input from the digital receiver by means of a dedicated 10 Gbit Ethernet line. The receiver output data rate is tunable from 0.5 to 255 MS/s ( $10^6$  samples/second) and the native format for each channel is a complex 8+8 bit signed integer (real ad imaginary parts as signed chars).

The desired spectral resolution is from 1 to 10 m/s. At a wavelength of 1 cm this translates to about 1 KHz. With a design band of 100MHz, this requirement translates in a 64K Fourier Transform.

To limit the spectral leakage a windowing (multiply of data with a special *windowing function*) must be applied before the Fourier Transform. The *windowing function* can be chosen to enhance different characteristics of the spectra.

The integration can be accomplished by means of a simple accumulation of the moduli of the different channels of the Transform.

## 2 Implementation details

The data source is connected to SpectralGpu by means of a fast 10GHz Ethernet link and we use a dedicated driver to feed data to the GPU engine. The network board is a Myricom Myri 10G-PCIE-8B-S+E, using a PCI Express X8 bus, with an expected throughput of 300MSamples/s (600 MB/s bandwidth). The actual maximum bandwidth, due to memory bottleneck, is around 400MHz. Full speed can be achieved by using a custom driver.

During the design of the SpectralGpu project we choose to use coprocessor GPU boards which have many simple elementary CPUs, as those manufactured by ATI. We evaluated that our problem involving many simple computation is best suited to simpler and more numerous engines.

That choice implies the use of OpenCL programming environment, with a much steeper learning curve in comparison with NVIDIA's CUDA. At the end, however, we got a finer grain control over the coprocessor. After a somehow long start-up phase, development proceeded fast and the GPU portion exceeded the planned speed during the firsts iteration cycles, marking a success in this milestone.

We have successfully tested our code on a NVIDIA 8300, using the OpenCL portion of CUDA SDK. As expected this combination has a poor performance, compared with the ATI boards.

Data are received by program as couples of 8 bit signed integers (real and imaginary part) and are written into the global memory of the GPU in blocks of 32KSamples.

On every iteration a fixed number of blocks are loaded and processed in parallel, and this number is chosen to maximize the global throughput. The best number depends on GPU architectures: we got the maximum computational speed processing from 128 to 512 blocks at time.

Each iteration activates four GPU kernels (Internal tasks):

**The Reader** Data are converted to floating point format (single or double), and convolved with the windowing function.

- **The radix-n FFT** Data are partially transformed to frequency domain. To get the Fourier Transform of the input this kernel is called a number of times which is a function of block's length and of the FFT radix. The FFT kernel operates on an input buffer and stores results in an output buffer. In the next iteration is then called with swapped input and output. The swap of buffers is done by the **swapper** kernel.
- The Swapper This kernel copies the output buffer of the FFT stage to the input buffer.
- **The Writer** The modulus of each spectra is averaged channel by channel, and then transferred to the host. Averaging each iteration result gives the *integrated* spectral power.

## **3** Performances

The SpectralGpu program part has been tested on different combination of hosts and GPU boards We present here only preliminary results. We plan to perform a test on a single host with different combination of drivers and boards. We present computational speed for both single and double precision. We also performed a computation on simulated, but realistic data, and we found that the rounding error was well below data noise even in the single precision case.

In the following table we list only the best results for each GPU:

GPU board	CPU	Driver version	Buffer lenght and	Max Speed
			number of buffers	MSample/s
ATI 7950 3GB	Intel Core i5-2400	12.6	$32 \text{KB} \times 512$	225MS/s Double
			$32 \text{KB} \times 512$	370MS/s Single
ATI 6950 1GB	AMD Athlon II	11.6	$32 \text{KB} \times 256$	200MS/s Double
	X4 630		$32 \text{KB} \times 512$	310MS/s Single
ATI 7770 1GB	Intel CoreDuo	12.6	$32 \text{KB} \times 128$	60MS/s Double
	Quad CPU Q6600		$32 \text{KB} \times 512$	160MS/s Single



Figure 1: SpectralGpu Logo

Version 1.0, November 2012.