Polyphase Filter Bank implementation in GWB

Version 1, June 2020

S. Harshavardhan Reddy

Polyphase Filter Bank :

Discrete Fourier Transform (DFT) suffers from two drawbacks i.e; leakage and scalloping loss. Applying Polyphase Filter Bank (PFB) supresses the drawbacks of DFT. In signal processing terms, PFB is a linear filter applied on frequency channels representing the DFT frequency bins after DFT. Another important use of PFB is that it serves as a band pass filter where desired frequency range is kept and the rest is made to zero.

Implementation :

Instead of taking a N-point FFT directly, a block of data of size N x P = M (where P represents number of taps) is read and multiplied point-by-point with a window function. The window function is basically a sinc function which is tranform of rectangular function as the aim is to have single bin frequency response to be rectangular. This window function can further be multiplied by any smoothing window function like hamming or hanning window for better response. After multiplication, the block of data is split into P subsets of length N each, and added point-by-point. Then an N-point FFT is performed on this array of data. Pictorial representation is given below.

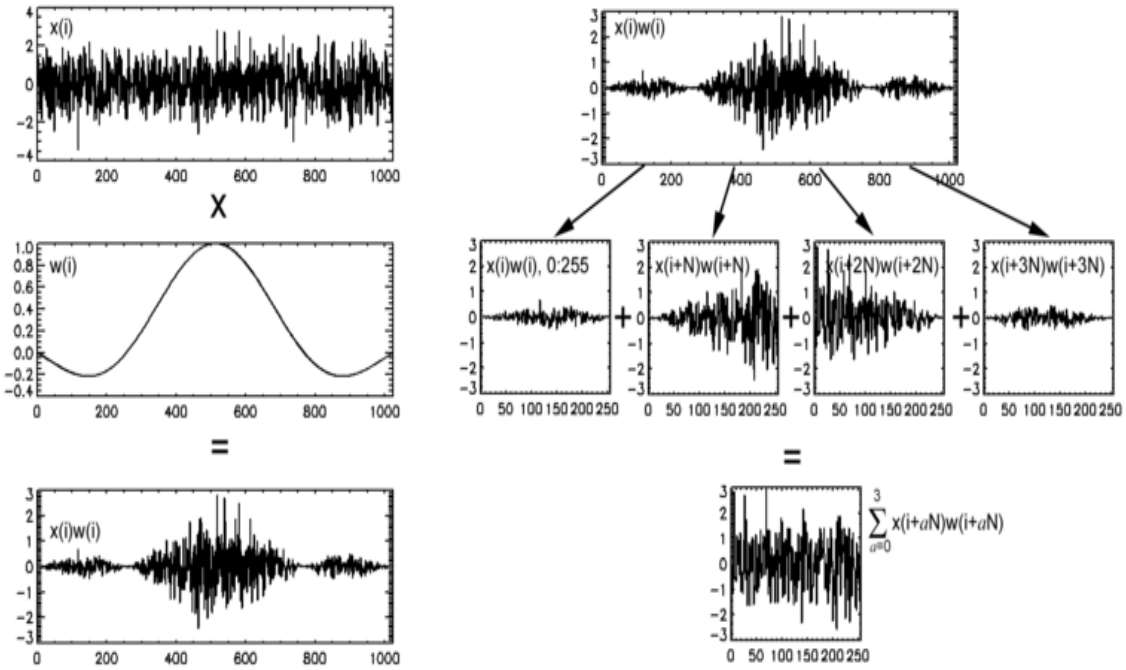

Fig. 1 : Graphical depiction of polyphase filtering. x(i) is a time series of length M = 1024 samples, multiplied point-by-point with the window function w(i) (a sinc function), also of the same length. The product is split into P = 4 blocks of length N = 256 samples each, and summed. This summed array of length N = 256 samples, shown at the bottom, on the right, is then input to a routine that takes a 256-point Fourier Transform. (Pic courtesy : "The Polyphase Filter Bank Technique" by Jayanth Chennamangalam)

Implementation in the GWB :

The GWB is implemented using time-slicing design i.e; the data stream from all antennas is sliced in time and each compute node with GPUs processes a slice of data stream from all antennas. The steps of processing in GPUs are converting 8-bit or 4-bit data into floating point followed by FFT using CUFFT library and multiplication and accumulation (MAC).

In the GWB, PFB is implemented after the stage where data is converted from 8-bit or 4-bit to floating point. For a slice of data, the last N (FFT size N) samples requires (P – 1) x N samples of next slice for performing the PFB operation. Hence, at the stage of time slicing and data sharing, extra data required for last N samples' PFB operation is also shared. In the narrowband mode case, PFB is applied on the decimated data.

Coming to the actual implementation in the GPU kernel, the window function is pre-calculated depending on the number of spectral channels and number of taps and stored in the global memory of GPU. In the kernel, each thread perform PFB (filtering) operation for a single sample. The number of threads in a block is kept fixed at 256 (Block Size). The block dimensions are then calculated such that, in the x-dimension the number of blocks = (Total number of samples per time slice per antenna / Block Size) and y-dimension the number of blocks = number of antennas * number of pols.

Computation requirements and performance :

On each time sample, the computation performed is equal to number of taps multiplication plus number of taps additions. If each multiplication takes one floating point operation and addition takes one floating point operation then for each sample the computation performed is (number of taps x 2) floating point operations. The data rate per polarisation per second is 400 MSamples/second. So, total computation required is (Bandwidth x 2 x number of taps x number of antennas x number of pols x 2). For bandwidth of 200 MHz and 8 taps, the total computation required for PFB is 409.6 GFlops. The performance achieved in K40 GPUs was around 75 GFlops. The relatively low performance compared to the peak performance of K40 GPU is because of non-coalesced memory access of global memory.

Test results :

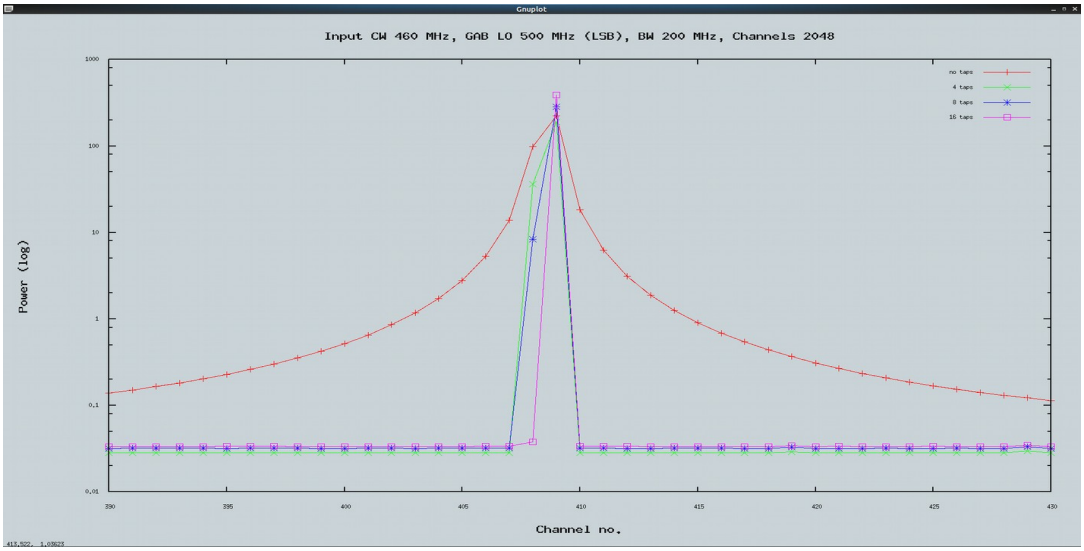

Fig. 2 : Plot showing effect of PFB at various tap lengths. BW : 200 MHz, CW Signal : 460 MHz, LO : 500 MHz, No. of channels : 2048

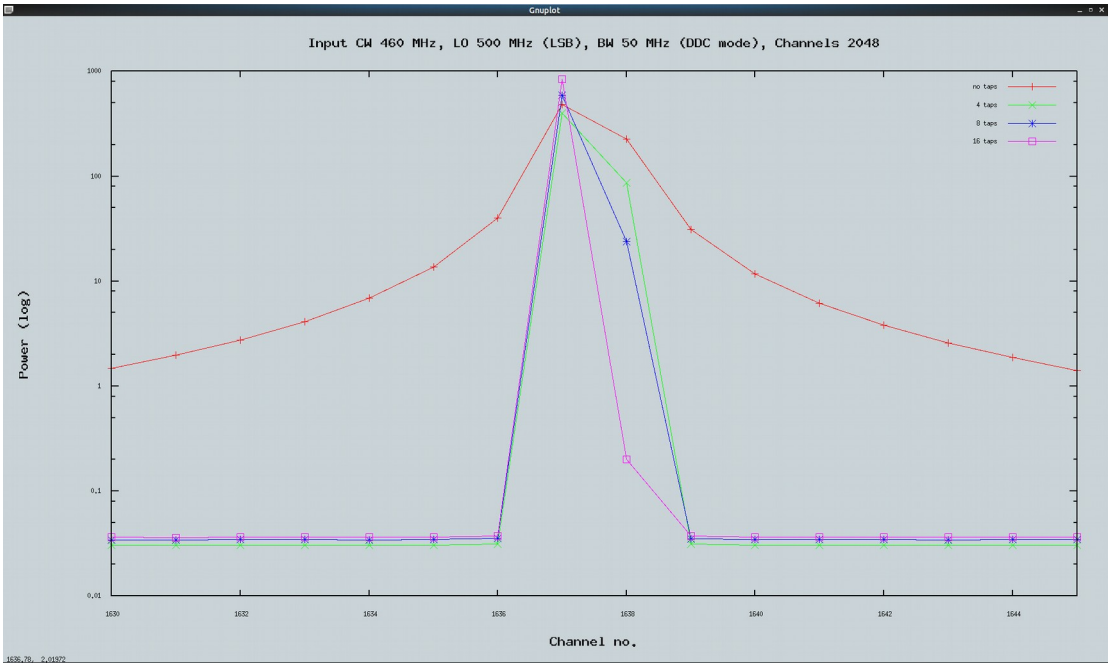

Fig. 3 : Plot showing effect of PFB at various tap lengths. BW : 50 MHz, CW Signal : 460 MHz, LO : 500 MHz, No. of channels : 2048

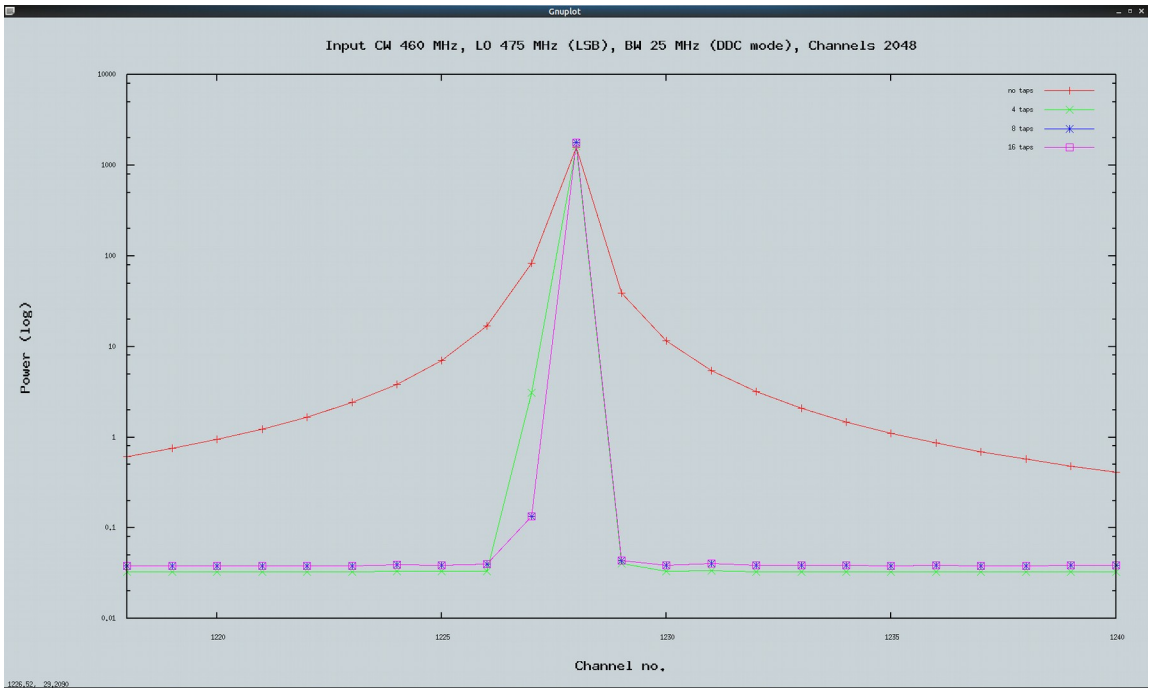

Fig. 4 : Plot showing effect of PFB at various tap lengths. BW : 50 MHz, CW Signal : 460 MHz, LO : 475 MHz, No. of channels : 2048

Possible modes in GWB with PFB :

As PFB comes at the cost of extra computation some modes of GWB where the computation time is near to real-time will not work as the total computation time including PFB will overflow the real-time. With the computation performance achieved on K40 GPUs, possible modes are listed below :

<table>
<tr><th>Bandwidth</th><th>Interferometry</th><th>Beamformer</th></tr>
<tr><td>200 MHz/ 100 MHz</td><td>Maximum taps = 16</td><td>All modes possible</td></tr>
<tr><td rowspan="2">400 MHz</td><td rowspan="2">Maximum taps = 4</td><td>Above 8192 channels no beams possible</td></tr>
<tr><td>Up to 8192 channels all modes possible</td></tr>
<tr><td rowspan="4">Narrowband mode</td><td>Decimation <= 4
Maximum taps = 16</td><td rowspan="4">All modes modes</td></tr>
<tr><td>Decimation = 8
Maximum taps = 8</td></tr>
<tr><td>Decimation = 16
Full Stokes mode, Maximum taps = 8
Total Intensity, Maximum taps = 4</td></tr>
<tr><td>Decimation > 16,
PFB mode not possible</td></tr>
</table>

References :

1. Chennamangalam J., [2011] The Polyphase Filter Bank Technique, http://casper.berkeley.edu/wiki/The Polyphase Filter Bank Technique.