MARK 5 MEMO #018

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
HAYSTACK OBSERVATORY
WESTFORD, MASSACHUSETTS 01886

November 19, 2004

Telephone: 978-692-4764
Fax: 781-981-0590

To: Mark 5 Development Group

From: Alan E.E. Rogers and Hans F. Hinteregger

Subject: Real to real polyphase filter bank

A] Method using overlapped FFTs

The following algorithms have been tested as a method of converting an I.F. from 16 to 512 MHz to 31 real 16 MHz USB channels:

1] Sample the input data with a 1024 MHz clock converting the 8 to 504 or 520 to 1016 MHz I.F. band from analog to digital.

2] Put the samples $x[t]$ in blocks of 64 samples into a series of 64 m tap FIR subfilters so that the output of the nth subfilter is

$$y_n[b] = \sum_{k=0}^{m-1} x[n + 64k + 64b]\, h[n + 64k]$$

where n = sub-filter number
b = data block number

and into another set of subfilters with a delay of 32 so that

$$z_n(b) = \sum_{k=0}^{m-1} x[n + 64k + 32 + 64b]\, h[n + 64k]$$

3] Move the y outputs into the real inputs of a 64 point complex FFT and the z outputs into the complex inputs

$$\mathrm{Re}\, X[n] = y_n[b]$$
$$\mathrm{Im}\, X[n] = z_n[b]$$

This allows the FFT to transform 2 real signals with one complex FFT.

4] For block b=0 take 64 point FFT

5] Combine the outputs Y[ ] of the FFT to separate the transforms of the first 64 samples put in the real inputs from those put into the imaginary inputs

$$\begin{aligned}
&\mathrm{Re}\, Z[0] = \mathrm{Re}\, Y[0] + \mathrm{Re}\, Y[0] \\
&\mathrm{Re}\, Z[n] = \mathrm{Re}\, Y[n] + \mathrm{Re}\, Y[64-n] \quad \text{for } n > 0 \\
&\mathrm{Im}\, Z[0] = \mathrm{Im}\, Y[0] - \mathrm{Im}\, Y[0] \\
&\mathrm{Im}\, Z[n] = \mathrm{Im}\, Y[n] - \mathrm{Im}\, Y[64-n] \quad \text{for } n > 0 \\
&\textit{and} \\
&\mathrm{Re}\, Z'[0] = \mathrm{Im}\, Y[0] + \mathrm{Im}\, Y[0] \\
&\mathrm{Re}\, Z'[n] = \mathrm{Im}\, Y[n] + \mathrm{Im}\, Y[64-n] \quad \text{for } n > 0 \\
&\mathrm{Im}\, Z'[0] = \mathrm{Re}\, Y[0] - \mathrm{Re}\, Y[0] \\
&\mathrm{Im}\, Z'[n] = \mathrm{Re}\, Y[64-n] - \mathrm{Re}\, Y[n] \quad \text{for } n > 0
\end{aligned}$$

6] Form the real output sequences for the 31 output channels as follows:

$$\begin{aligned}
&C_n[2b] = s_0\, \mathrm{Re}\, Z_b[n] \quad \textit{for } 0 < n < 32 \\
&C_n[2b+1] = s_0 s_1\, \mathrm{Im}\, Z'_b[n]
\end{aligned}$$

where $s_0 = 1$ for even blocks
$s_0 = -1$ for odd blocks
$s_1 = 1$ for even channels
$s_1 = -1$ for odd channels

The n=0 channel is not useful due to aliasing.

7] Form the next set of samples for each channel from the next block b=1 etc.

The FIR filter coefficients can be obtained from sampling a windowed sinc function as follows:

$$h[i] = 0.54 - 0.46\cos\left(2\pi i/(N-1)\right)\mathrm{sinc}(w)$$

where $w = (i - N/2)\pi/64$
$N = m \times 64$

B] Method using non overlapped FFTs

1] Sample input with 1024 MHz clock

2] Put samples into FIR subfilter

$$y_n[b] = \sum_{m=0}^{m-1} x[n + 64k + 128b]h[n + 64k]$$

and

$$z_n(b) = \sum_{m=0}^{m-1} x[n + 64k + 64 + 128b]h[n + 64k]$$

3] Put into 64 point FFT

$$\text{Re}\, X[n] = y_n[b]$$
$$\text{Im}\, X[n] = z_n[b]$$

4] Take FFT

5] Combine outputs to separate results

$$\text{Re}\, Z[n] = \text{Re}\, Y[n] + \text{Re}\, Y[64 - n]$$
$$\text{Im}\, Z[n] = \text{Im}\, Y[n] - \text{Im}\, Y[64 - n]$$
$$\text{Re}\, Z'[n] = \text{Im}\, Y[n] + \text{Im}\, Y[64 - n]$$
$$\text{Im}\, Z'[n] = \text{Re}\, Y[64 - n] - \text{Re}\, Y[n]$$

6] Form output sequence of 4 samples per 128 sample block for each channel

$$C_n[4b] = \text{Re}\, Z_b[n]$$
$$C_n[4b + 1] = I[2b]$$
$$C_n[4b + 2] = -\text{Re}\, Z'_b[n]$$
$$C_n[4b + 3] = -I[2b + 1]$$

where I [ ] are interpolated values from $\text{Im}\, Z$ and $\text{Im}\, Z'$ formed by taking the sequence Q[k] and applying a FIR interpolation filter

$$Q[b] = \text{Im}\, Z_b$$
$$Q[2b + 1] = \text{Im}\, Z'_b$$

$$I[b] = \sum_{0}^{m-1} Q[b + k - m/2]H[k]$$

The interpolation FIR has values of $H[i] = \left(0.7 - 0.3\cos\left(2\pi\, i/(N-1)\right)\right)\text{sinc}\left((i - N/2 + 0.5)\pi\right)$

16 taps are sufficient to provide more than 40 dB image rejection across 80% of each filter output.

C] Method using mixing before the FFT

An alternate method is to perform SSB mixing operations prior to entering the FFT. Following the method of Crochiere and Rabiner's figure 7.48:

1] Sample the input with A/D

2] Put the samples $x[t]$ into a series of 32 m tap complex subfilters

$$y_n[b] = \sum_{k=0}^{m/2-1} x[n+64k+32b]\,h[n+64k]\,s_0$$

$$z_n[b] = \sum_{k=0}^{m/2-1} x[n+64k+32b+32]\,h[n+64k+32]\,s_0$$

where s_0 = 1, -1, 1, -1 etc.

in this case

$$h[i] = 0.54 - 0.46\cos\left(2\pi i/(N-1)\right)\mathrm{sinc}(w)$$

where $w = (i - N/2)\,\pi/64$

$N = 32m$

3] Perform a complex rotation of each output of FIR sub-filters before putting into the FFT.

$$X_b[n] = \left(y_n[b] + iz_n[b]\right)e^{i\frac{2\pi n}{128}}$$

4] The real outputs of the FFT form the real samples for each channel. The output channel order in order of increasing frequency without resorting will be

0, 31,1, 30,2, 29, 3, 28, 4, 27, 5, 26, 6, 25, 7, 24, 8, 23, 9, 22, 10, 21, 11, 20, 12, 19, 13, 18, 14, 17, 15, 16

and after resorting the channels alternate between USB and LSB. Figure 1 shows the 32 output filter responses from the second method using windowed sinc symmetric FIR. [More work will be needed to optimize the filter design.] The bandpasss function for the methods A&B are similar and are shown in Figure 2. The first channel is not useful but since this channel is at the bandpass end it is likely that it would be unusable anyway owing to the shape of the analog bandpass filter ahead of the A/D.

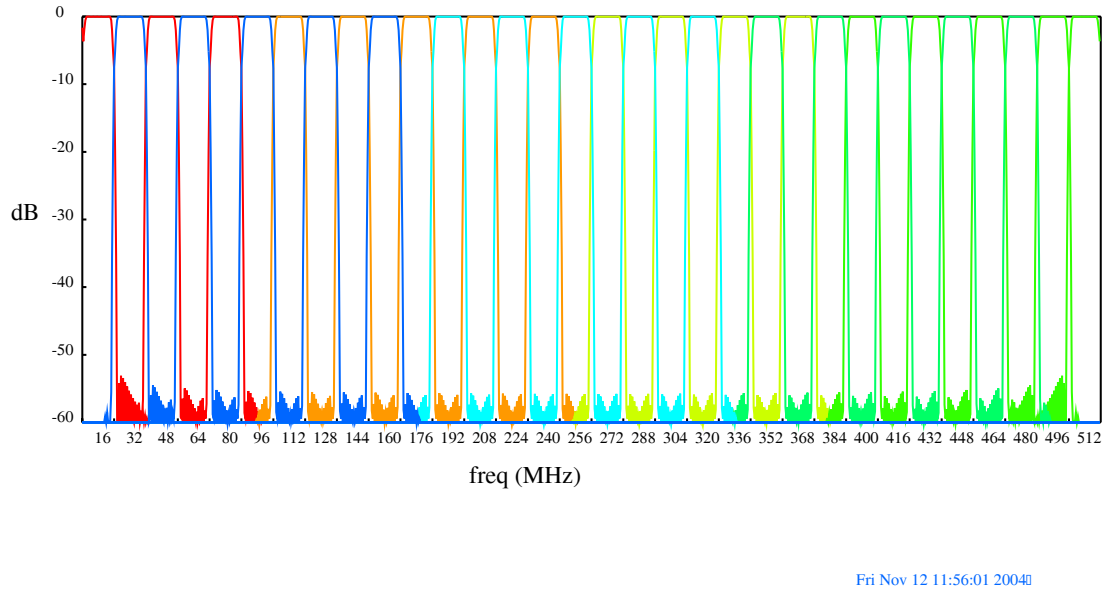

Figure 1. Channel bandpasses from method C

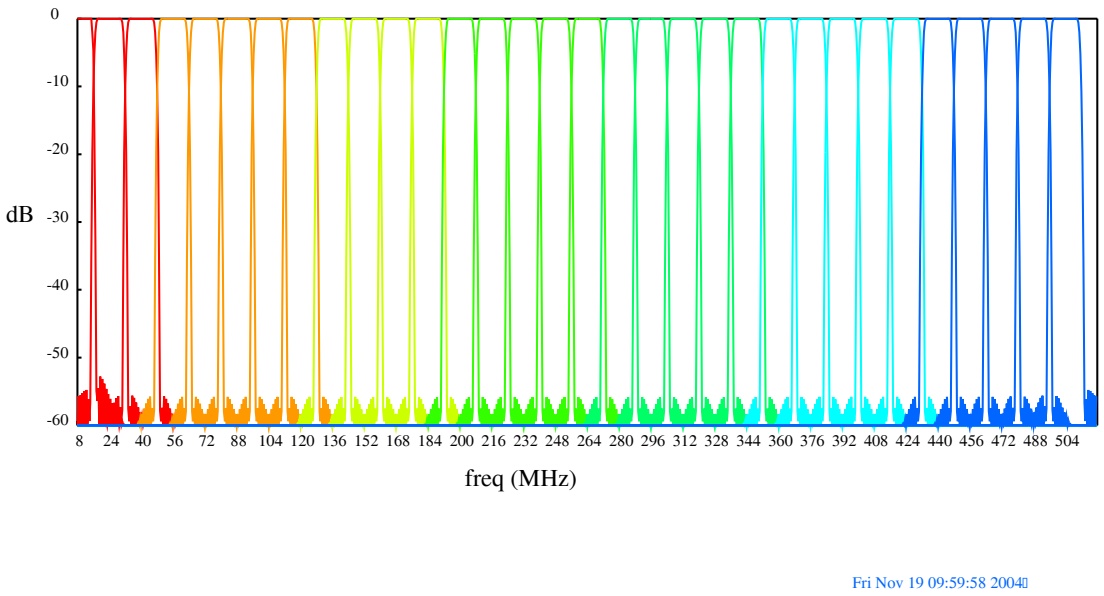

Figure 2. Channel bandpass from methods A and B

Comparison of computation required for 3 methods:

A.) Real to real filter bank using overlapped FFTs
This is the first method described in this memo. It uses overlapped FFTs to provide interpolated outputs so that the complex outputs can be shifted to positive frequencys by multiplication by a vector that rotates by 90 degrees per output sample and subsequent dropping of the imaginary part. For efficiency one 64 point FFT serves double duty by processing both the primary and the overlapped FFTs.

B.) Real to real filter bank using interpolation of outputs
This method is similar to method A except that the FFTs are not overlapped so that 1 "double duty" 64 point FFT services 128 input samples. Every other output sample needs to be time shifted by interpolation

C.) Real to real using SSB mixing prior to FFT
This is the third method described in this memo. For the details of this method we refer to Crochiere and Rabiner. This method has the advantage of providing all 32 filtered outputs. The following table compares the number of multiplications needed for each algorithm. This real to real method requires about twice the number of multiplications needed by the complex to complex method.

D.) Complex to complex filter bank
The simplest polyphase filter bank is one which converts complex samples input to complex outputs. A 32 point FFT is needed to obtain 32 filtered outputs. We take this as our "reference" method. However for VLBI we require real outputs.

					Multiplies per 64 ns				Input Subfilter	Output filters
	I/O	Overlap FFT	Sample rate Gs/s	FFT size	FFT	FIR PFB	FIR INTR	Tot	Size m	Size m
A	Real/real	Yes	1	64	768	2048	-	2816	16	None
B	Real/real	None	1	64	384	1024	256	1664	16	16
C	Real/real	None	1	32	640	2304	-	2944	32	None
D	Cmpx/cmpx	None	0.5	32	320	1024	-	1344	16	None

Table. Polyphase filter bank for 32 output channels for 500 MHz bandwidth

Notes:

1) Assumes $2n\log_2 n$ multiplies for an n point FFT
2) The FIR, assumed for the complex filter bank these calculations has 16 taps per sub-filter or a total of 512 taps. The FIRs for the real to real cases are chosen to have the same shape factors as the complex to complex case.
3) Method B is similar to method A but there is no overlap so that 1 64 point FFT services 128 input samples. The extra output samples required to allow conversion to real are obtained using a FIR filter on each output to interpolate between output samples. Only the imaginary outputs need to be interpolated and the filter is symmetric so that only half the number of multipliers are needed.