RRI Internal Technical Report – Library_ref_no. *TR: AA-2007-4*

Grant Hampson's polyphase filter bank and its implementation in the ADFB firmware

Authors: D. Anish Roshi[1], K. S. Srivani[2]
Affiliation: [1,2] Raman Research Institute, Bangalore, India

Date: October 1, 2007
Class: MWA: Receiver Node

Abstract

The report presents the characteristics of Grant Hampson's polyphase filter bank (PFB) and suggested implementation for MWA application. Grant's PFB will be used in the version 1.0 firmware that is being developed for the Analog-to-digital converter (ADFB) board of the MWA.

1 Introduction

The Analog-to-digital converter (ADFB[1]) board consists of dual 8 bit ADCs (AT84AD001B) followed by field programmable gate arrays (FPGA; V4SX35). The 330 MHz analog bandwidth will be Nyquist sampled at 660 MHz in the ADFB board (see Roshi et al. 2007 for details). In version 1.0, a PFB will break the 330 MHz bandwidth into 256 channels and 24 channels will be send to the backend processing unit. This PFB will be implemented in the FPGAs in the ADFB board. We plan to use the PFB firmware developed by Grant Hampson at ATNF in version 1.0. This firmware is provided to MWA as an EDIF file. We used the Xilinx synthesis software ISE 8.2 and Modelsim 5.8b to simulate and study the performance of the PFB in order to integrate it with the rest of the design. This report summarizes our understanding of the PFB. The results presented here are based on the functional simulation of the PFB using the above mentioned software version with updated Xilinix core library from ISE 9.2.

2 Grant's PFB

A simplified block diagram of the PFB provided by Grant for the MWA work is shown in Fig. 1. The input data samples are divided into 4 sequences and are 9 bits wide two's complement format. The PFB provides two 16 bit complex outputs consisting of 257 values corresponding to channels 0 to 256. The mapping of output data sequence to channel number is given in Table. 1. The filter co-efficient and trig values are 12 bits wide.

3 Sine wave input to the PFB

Sine wave centered at channel 10 was fed to the PFB. The amplitude of the sine wave is quantized to 9-bit two's complement format. A plot of the ratio of the input to output SNR (signal-to-noise ratio) as a function of sine wave amplitude (see Fig. 2) shows that the PFB performs well up to a

[1]This board is made based on a design provided by CSIRO-ICT center, Sydney

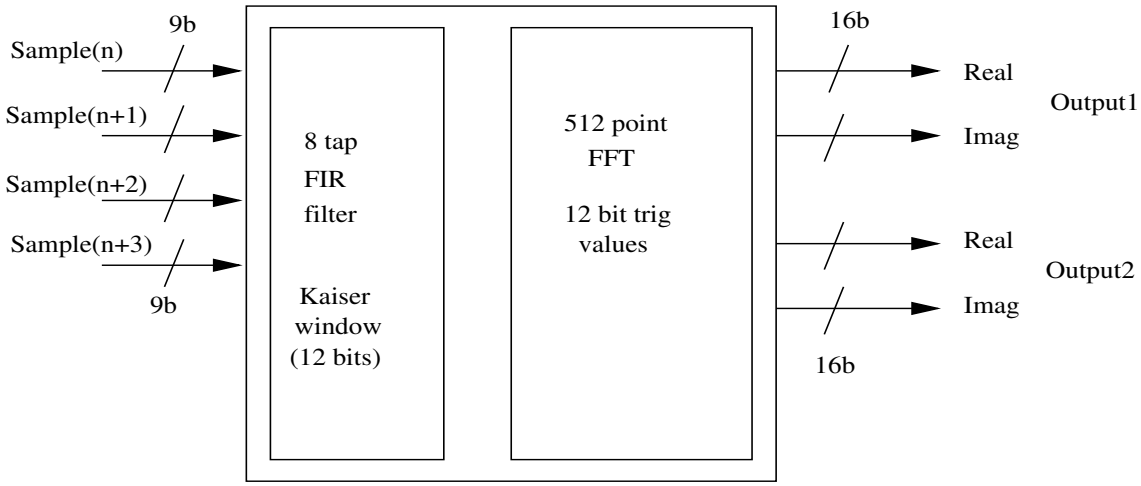

Figure 1: Block diagram of Grant's PFB

Table 1: Mapping of the PFB output sequence to channel number

Seq.	Output1		Output2	
No.[a]	Real	Imag	Real	Imag
1	Rch0[b]	Rch256	Rch128	Ich128
2	Rch1	Ich1	Rch255	Ich255
3	Rch2	Ich2	Rch254	Ich254
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
127	Rch126	Ich126	Rch130	Ich130
128	Rch127	Ich127	Rch129	Ich129

Note: (a)Sequence number corresponding to the PFB outputs.
(b) Rch$i$ and Ich$i$ correspond to the real and imaginary values of the i^{th} channel respectively. ch0 and ch256 are respectively the DC and Nyquist channel.

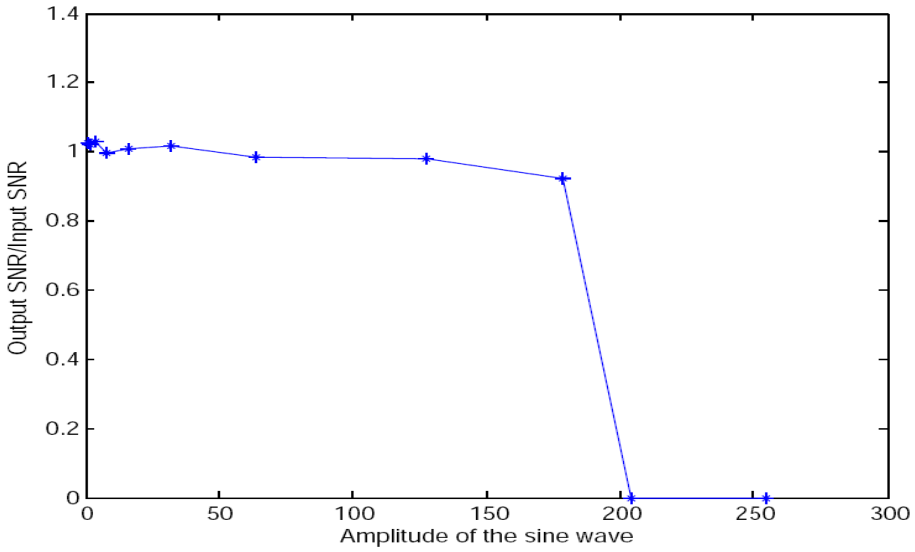

Figure 2: Ratio of the input to output SNR vs input sine wave amplitude. The SNR (signal-to-noise ratio) is estimated as the ratio of the spectral power at channel 10 to the mean power in other channels. For this measurement a sine wave is fed to the input such that the delta function is located at channel 10.

maximum input amplitude ($\sim$ 180 in decimal). Here SNR is estimated as the ratio of the spectral power at channel 10 to the mean power in other channels. Fig. 3 compares the input spectrum, obtained with double precession FFT, with the output spectrum for different sine wave amplitude. This plot shows that the PFB reproduces the spectrum without any further degradation as long as the input amplitude does not the maximum value of 180. Examining the PFB output numbers indicates that the restriction in the input amplitude is due to overflow of the spectral values in the 16 bit output representation.

4 PFB output quantization to (5,5) bits

Constraints on data transport through fiber from the Receiver Node require that the number of bits for representing the real and imaginary parts be reduced to 5 bits. We simulate the bit reduction implementation at the output of the PFB by truncation and rounding-off for different inputs. Results for multiple sine waves with constant power gain across the band and with a gain function are shown in Figs 4 & 5. Based on these simulation we propose to implement the rounding-off scheme with gain correction for the MWA (see Section 5).

Simulations were also made for Gaussian noise with a gain function and the results are shown in Figs 6. For this simulation, a Gaussian noise is convolved with a 7th order FIR filter response. The power spectrum of the convolved noise is shown in Figs 6a. The RMS value of the input sequence is kept at 1/8 of the full range of the ADC. A voltage gain correction function estimated from the power spectrum by taking its square root and normalizing to the peak value (see Figs 6b) is used to correct the gain variation across the frequency. The mean power after gain correction is used to get an estimate of the RMS value ($= \sigma$) of real and imaginary outputs of the PFB. The range of real and imaginary values are then taken as $\pm 3\sigma$ level. The most signification 5 bits of this range were taken for the final output after rounding-off the values. The average power spectrum

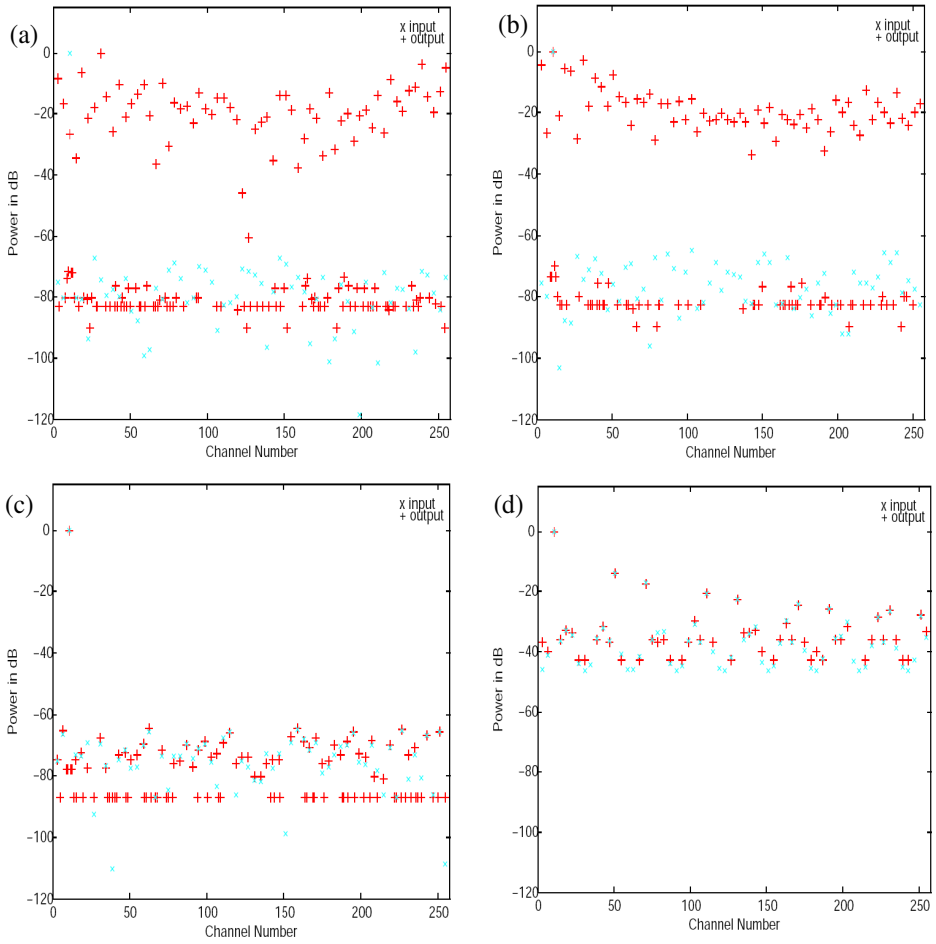

Figure 3: Plots showing the comparison of the input spectrum (cross) obtained with double precision FFT with the output (plus) of the PFB. Amplitudes of the sine waves in decimal values are 255, 204, 178 and 1 for the plots in (a),(b),(c) and (d) respectively.

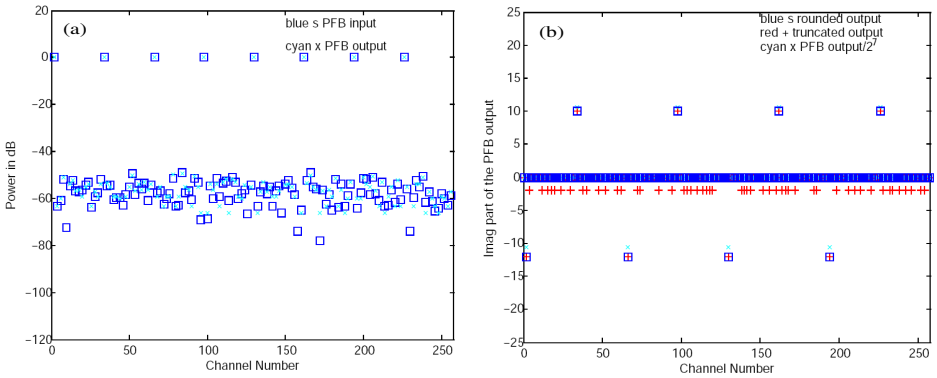

Figure 4: (a) Plot shows the input spectrum (square), obtained with double precision FFT, along with the output (cross) of the PFB. The input in this case is a sum of 8 sine waves quantized to 8 bits. The bit length is extended to 9 bit by duplicating the 8th bit after conversion to two's complement form. (b) Plot shows the imaginary part of the PFB after converting to 5 bits. Bits 8 to 12 of the 16-bit output of the PFB are used for the conversion. The result of truncating to 5 bits is shown with cross, where the 1 bit error is clearly seen near 0 values. The output of the PFB after rounding-off to 5 bits is shown with squares. For comparison the original 16-bit values are scale by 2^7 and shown with cross in the plot.

obtained with the 5-bit values compares well with the full precision spectrum after appropriate scaling (see Figs 6c). Fig. 6d shows the difference between the 5-bit quantized real values and the corresponding 16-bit PFB output; the 16-bit output is scaling by 2^7 since bits 8 to 12 are used for the 5-bit quantization.

5 PFB implementation for ADFB version 1.0 firmware

A simplified block diagram of the implementation of PFB in the version 1.0 ADFB firmware is shown in Fig. 7. Note that the details of the actual implementation are omitted and will be presented elsewhere. The ADC used in the ADFB board is an 8-bit converter and we plan to use the binary output mode. In this mode, the most negative and positive voltages are represented by 0x00 and 0xFF respectively. Thus for Walsh switching a NOT operation has to be done (see Fig. 7). The PFB input is in 9 bit two's complement format. We plan to use a look-up table for the conversion of 8 bits ADC data (as well as for Walsh switching) and the 9th bit input of the PFB will be tied to the 8 bit. This will allow us to use the full range of the ADC.

The 16-bit output of the PFB will be scale with 'real' voltage gain values, quantized to 16 bits, for each channel and then round-off to 5 bits. The DSP48 slice in Virtex 4 can implement this scaling and rounding-off function.

Acknowledgment

We thank Grant Hampson for providing his PFB code, which will be used in version 1.0 of the ADFB board.

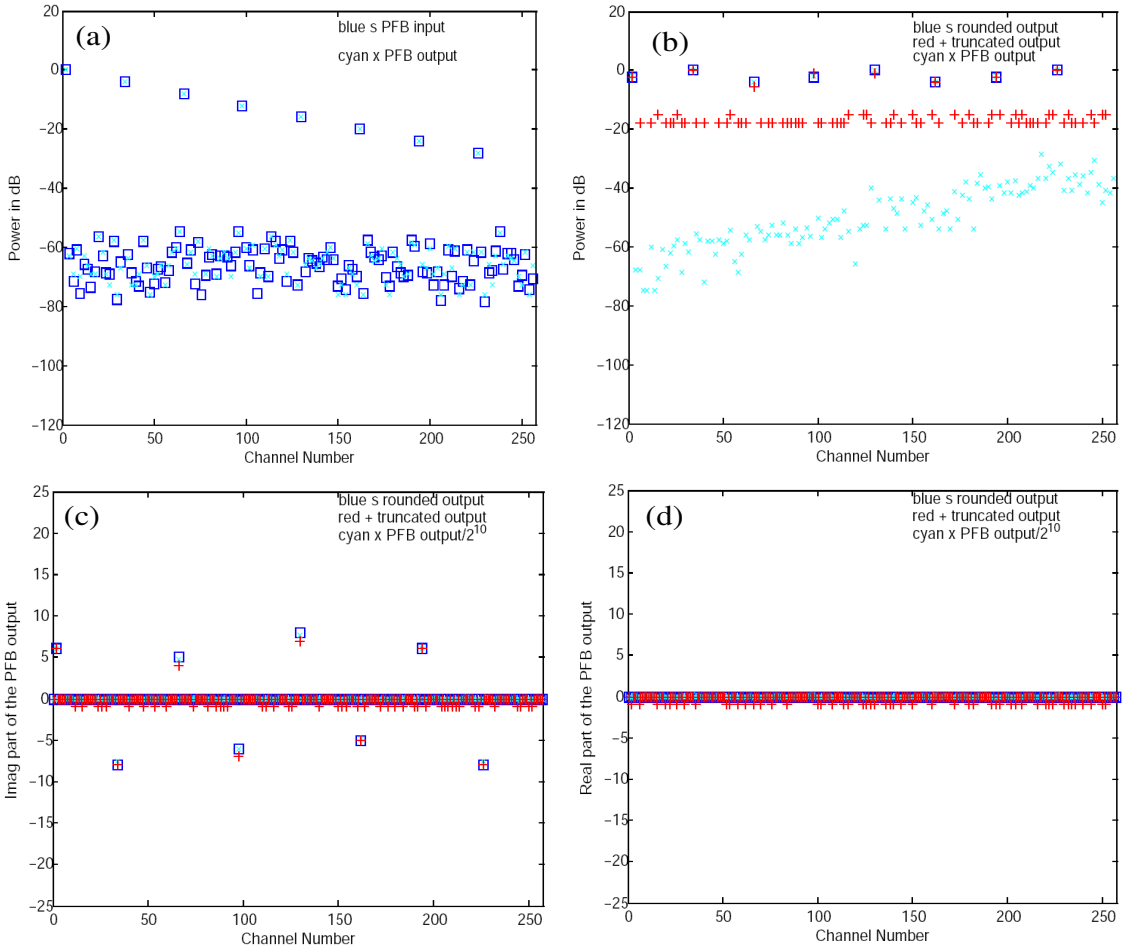

Figure 5: (a) Plot shows the input spectrum (square), obtained with double precision FFT, along with the output (cross) of the PFB. The input in this case is a sum of 8 sine waves of different amplitude quantized to 8 bits. The bit length is extended to 9 bit by duplicating the 8th bit after conversion to two's complement form. The power of the sine wave components varies by about 24 dB between the lowest and highest channel number. This variation in power can be considered as caused by the power gain of the analog receiver system (b) Plot of the input spectrum and the PFB output after correcting the gain variation. The increase in the quantization noise power with channel number is due to the gain correction. The spectrum after truncating the 16-bit PFB output to 5 bits is shown with cross and that after rounding-off to 5 bits is shown with square. The low level (about -20 dB) noise power in the quantized output is a result of the 1 bit error due to truncation. (c) and (d) Plot of the imaginary and real parts of the PFB output after truncating (cross) and rounding-off (square) to 5-bits. Quantization to 5 bits is done after scaling with the square root of the power gain. Bits 11 to 15 of the 16-bit output of the PFB were used for the conversion. For comparison the original 16-bit values were scale by 2^7 and shown with cross in the plot.

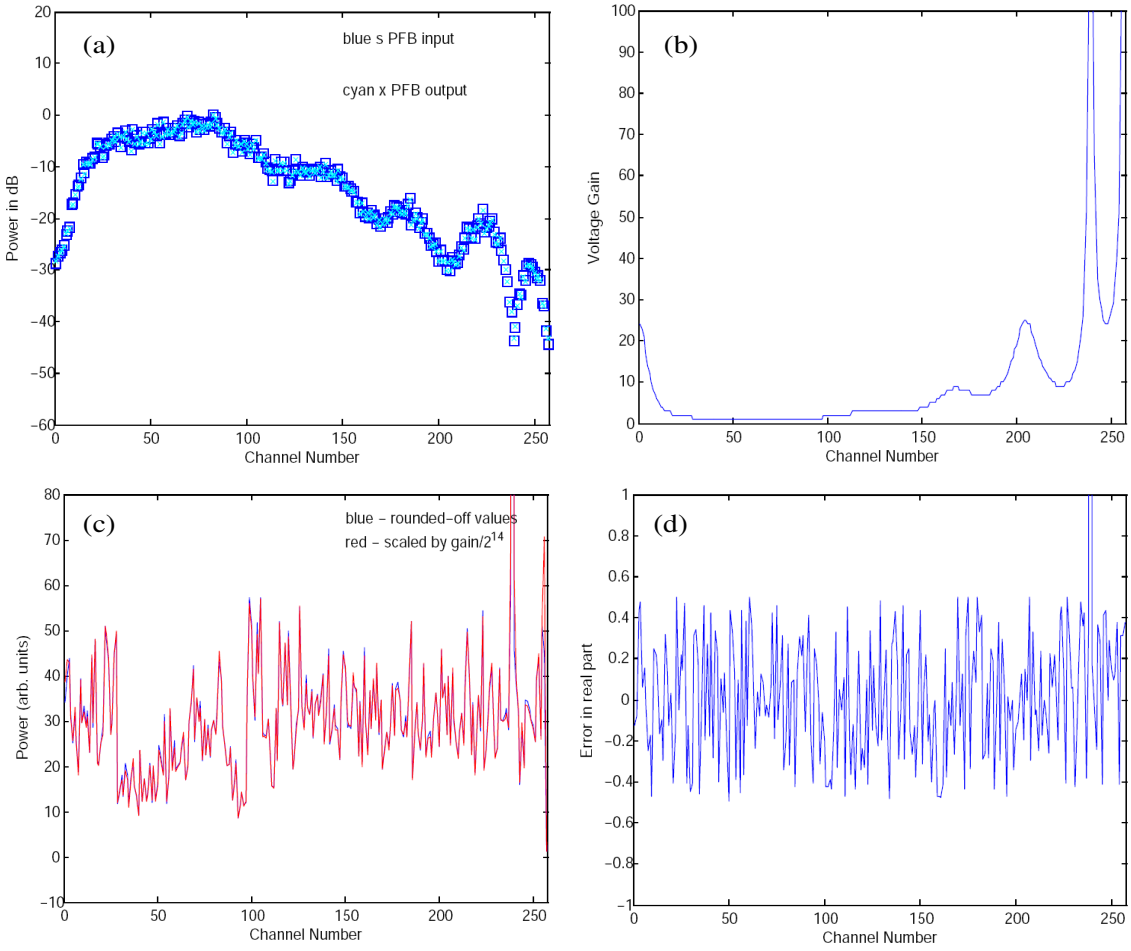

Figure 6: (a) Plot shows the average power spectrum (square) of the input Gaussian noise, obtained with double precision FFT, along with the output (cross) of the PFB. (b) Plot of the normalized voltage gain applied to correct the gain variation across the band. The gain values are quantized to 16 bits for the simulation. (c) Plot of the average spectrum (average of 450 spectra) after gain correction. The spectrum obtained after rounding-off the output values to 5-bits is shown in blue. For comparison the spectrum obtained from the original 16 bits values after correcting with the same gain function and scaling by 2^{14} is shown in red. (d) A typical difference between the 5-bit quantized and 16-bit values. 16-bit values are scaled by 2^7 before taking the difference. The error is distributed between $\pm$ 0.5.

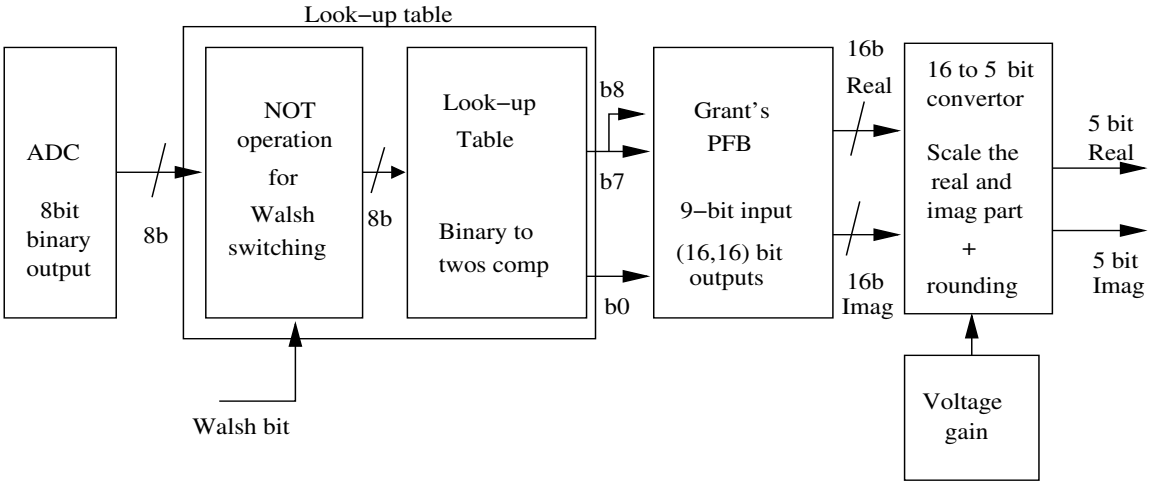

Figure 7: A simplified block diagram of the PFB implementation for version 1.0 ADFB firmware. Note that several associated functional blocks are not shown here and will be presented elsewhere.

Reference

Roshi, D. A, et. al., June 2007, MWA Knowledge Tree

Revision History

V1.0 – October 1, 2007