The pipelined frequency transform (PFT)

By John Lillington, RF Engines, Electronic Engineering
Jun 17, 2002 (3:58 AM)

This paper gives an outline description of the Pipelined Frequency Transform (PFT) hardware architecture, which was developed as an answer to the problem of channelising very wideband (80MHz+) signals. A number of commercially available solutions exist where a few narrowband channels need to be selected from a wideband channel, using conventional digital down-converter techniques. Likewise, by using pipelined FFT techniques, it is possible to completely “channelise” a wideband signal provided that only modest channel filter response is required. Where a requirement exists for a large number of channels and good filter performance, the above techniques become uneconomical. The PFT provides a much more optimum solution to this requirement.

General description of the PFT

The underlying concept is one of frequency band splitting where each successive stage of the PFT increases the number of bands by a factor of two, as shown in figure 1.

This band splitting could be achieved, for example, by a simple tree structure, as shown in figure 2. The input, which is complex to preserve positive and negative frequencies, is firstly split into two equal bands using a complex down-converter (CDC) and a complex up-converter (CUC). It would be possible to halve the sample rate for each of the sub-bands since the bandwidth of each has been halved. In practice, a degree of over-sampling is required to avoid image response problems caused by finite filter cut-off rates. Two times over-sampling is used at the output of the first stage. For all successive stages the output is decimated by two, preserving the overall two times over-sampling throughout the system.

The most obvious disadvantage of this approach is that, for large numbers of channels, the Tree gets impossibly large. For example, 1024 channels would require 2046 complex (CDC or CUC) modules.

Each one of these modules would take the form of figure 3 which shows, for example, the conventional form of the CDC(A) module, consisting of four multiplies, two adds/subtracts, a sine/cosine look-up table and a pair of low-pass filters.

The CUC(A) module would be very similar (differing only in the signs of the adder / subtractor elements). Successive stages would also be very similar except that the local oscillators are now at Fx/8 (where Fx is the input sampling rate for the stage) and the output is decimated-by-two.

Simplification of the architecture
Fortunately, the architecture can be greatly simplified in three significant ways. Firstly, with the tree system, the sampling rate drops by a factor of two at each stage. This would lead to inefficient use of the hardware which is capable of running at the full rate, Fs. The most processing-intensive part of each stage lies in the low-pass filters and, since these take an identical form within any given stage, interleaving techniques may be used to regain full efficiency. This involves interleaving the samples for each of the branches within a given stage and modifying the filters (which are normally finite-impulse-response or FIR filters) by adding extra delays between the coefficient multipliers. This is illustrated in figure 4.

The first stage complex converters, CDC(A) and CUC(A) are the same as for the Tree system (figure 2) but the subsequent stages, ICDC(B), ICUC(B) etc, are the Interleaved form of CDC(B), CUC(B) etc. Note that, after the interleavers, the sample rate at each stage is now constant at 2Fs, consistent with a two-times-oversampled system.

The second way in which greater efficiency may be obtained is to simplify the structure of each stage by eliminating unnecessary computations (eg, multiplying samples by local oscillator zeros). This is illustrated in figure 5 which shows how the architecture of CDC(A) can be greatly simplified.

Even further simplification can be achieved by combining common functions such as the sample interleavers and Local Oscillators. This is illustrated in figure 6 which shows how common functions of CDC(A) and CUC(A) may be combined.

Similar techniques may be used in subsequent stages, even though the Local Oscillator and interleaved filter structures are more complex.

The third way in which greater efficiency may be obtained is in the way coefficient multipliers are handled in the FIR filters. Firstly, the use of full multipliers (ie, multipliers where both inputs can be varied) may be avoided altogether and replaced by very efficient shift-and-add structures. Furthermore, since the Local Oscillators are generally in the form of +1 multipliers, these may be incorporated into the shift-and-add structure, forming simple polyphase filters.

There are other ways in which even greater efficiency has been achieved (including proprietary methods of realising adder trees and symmetrical rounding of data) which will not be covered in this paper.

Comparison of the PFT with FFT techniques

As mentioned earlier, FFT techniques, including Pipelined FFT may be used to produce a large number of equally spaced channels quite efficiently. A problem arises, however, when trying to achieve a large number of channels and good filter performance simultaneously.

This can be illustrated by referring to figure 7. A standard, unweighted FFT will have an effective filter performance as shown — ie, a simple SinX/X response. This means that the filter stop-band performance is poor (first sidelobe only 13.5dB down) and that there is roll-off within a "bin" (bin-width being defined as the Input Sample Rate Fs/Number...
of Bins).

By comparison, the PFT has a very flat frequency response across a bin and then cuts off very rapidly to the design stop-band level (approximately -85dB in this example). The degree of flatness and stop-band level are under the designers control using standard FIR filter design techniques.

The standard approach to improving stop-band performance with an FFT is to use weighting or "windowing" of the time-domain data. A number of standard windows exist including Hamming, Kaiser and Blackman-Harris.

An example using a Kaiser weighted window is shown in figure 8. Although this has clearly improved the stop-band performance it is still inadequate for many applications and has been gained at the expense of filter selectivity (ie, the main lobe is much wider).

The obvious question to ask is, can we not use FIR filter-like weightings on the FFT input data to achieve similar performance to the PFT? The answer is that it is possible provided much larger FFT's with overlap are used. The graph of figure 9 illustrates the point.

This graph determines the FFT requirement for performance equivalent to a 1024 bin PFT with given stop-band. For example, an 85dB stop-band could be achieved by using 5-off overlapped 5120 point weighted FFTs. The overlap needs to be such that a new 5120 point FFT is performed every 1024 points to ensure continuous real-time throughput, hence the need for 5-off parallel FFT's as demonstrated in figure 10.

Quite apart from the difficulty of actually estimating and realising the 5120 weighting coefficients, running such large mixed raddix FFTs at high throughput rates is a formidable task. Conservative estimates show that, even if such an architecture were feasible, it would require at least twice the amount of silicon needed for the equivalent PFT.

Comparison of the PFT

with a conventional digital down-converter

As mentioned in the introduction, standard digital down-converter (DDC) techniques are probably the best solution where only a few channels are required. This section attempts to compare the relative efficiency of the PFT and a stacked digital down-converter approach. In order to make a meaningful comparison, implementations need to be on the same platform. In this case the Xilinx Virtex-E platform is used since there is already a working implementation of the PFT for this. Furthermore, we can conveniently use Xilinx LogiCores to perform the various functions within the digital down-converter and give a reliable estimate of silicon usage.

Figure 11 shows a particular implementation of a single complex converter which would be one element of a 1024 channel system. It assumes an input sample rate of Fs=102.4 MHz at 8bits, complex and an output rate of 200 kHz at 16bits, complex. The output bandwidth is flat to +50kHz with a final filter shape corresponding to the overall 1024 bin, 85dB PFT response.

The filtering is based on a fairly standard combination of Cascaded Integrator Comb (CIC) and Decimating FIR filters which gives an efficient implementation. In this case, the CIC is a third order, decimate-by-64 (R=64) design and the FIR is a decimate-by-8 (S=8) design.

The CIC filter allows internal bit-growth to 27bits before rounding to 14bits and the FIR allows internal growth to 30bits before final rounding to 16bits. The Direct Digital Synthesiser (DDS) design allows for a minimum frequency increment of 50kHz (LUT depth of 1024) and an amplitude resolution of 14bits.

When considering the architecture for a multiple up/down-converter, it is obvious that greater efficiency can be achieved if the FIR filters can run at maximum rate.

Figure 12 shows how an N channel system may be split into N/R blocks where R is the CIC decimation. Taking an example using N=1024 channels, the CIC outputs will run at a decimated sample rate of 1.6MHz (R=64) and can be multiplexed in groups of 64 to regain the full rate of 102.4MHz. The FIR must now be an interleaving design to process the 64 interleaved data streams. This gives a small penalty to provide the extra inter-tap delays but this can be provided by the Virtex-E internal memory. To provide a block of 64 channels now needs 64 complex converters, 64 CICs but only one FIR.

To provide the full 1024 channels would require 16 such blocks. More generally, N channels would require N/R blocks.

What does this mean in terms of silicon usage? Table 1 shows a comparison for various PFT sizes from 2 to 1024 channels. For convenience, the comparison is made on number of slices using the Xilinx Virtex-E series. The PFT figures are based on actual working implementations and the Digital Down-Converter (DDC) figures are derived from five COREgen modules (DDS, Parallel Multiplier, Adder/Subtractor, CIC and Distributed Arithmetic FIR). The parameters of the COREgen modules were optimised as far as possible for each case (eg, reduced LUT length in the DDS modules for coarser frequency resolution for cases with lower number of bins).

A clearer comparison can be made in figure 13. At lower numbers of bins (up to around eight) there is little to choose between the two approaches.

The apparent advantage of the PFT, even at low numbers of bins is mostly to do with the high efficiency of the FIR filter design compared with the highly parameterised design for the COREgen modules. The differential could be reduced by using similar FIR designs in a customised DDC.

At higher number of bins, the advantage of the PFT becomes very obvious. A 1024 bin PFT at 26628 slices can fit comfortably into a Virtex XCV3200E (32448 slices). An equivalent DDC design (743040 slices) would require some 23 such devices.

Even at 128 bins, the DDC design would need more than two XCV3000E devices whereas the PFT can fit into an XCV1000E (12288 slices).

Figure 1: Frequency band splitting.
The pipelined frequency transform (PFT)

Figure 2: Simple tree system (3 stages)

Figure 3: Conventional form of complex down-converter (CDC(A)).

Figure 4: Block diagram of interleaved system (3 stages).

Figure 5: Simplified CDC(A) architecture.

Figure 6: Combined CDC(A) and CUC(A) architecture.

Figure 7: Comparison of typical PFT filter and unweighted FFT.

Figure 8: Comparison of typical PFT filter and Kaiser weighted FFT.

Figure 9: Comparison of PFT with multiple overlapped FFTs.

Figure 10: Pipelined FFT processing to generate a 1024 point output with 85dBc stop band rejection in real-time.

Figure 11: Implementation of single digital down-converter (DDC).

Figure 12: Architecture for multiple stacked DDCs with N bins.

Figure 13: Comparison of PFT and DDC implementations, 2 to 128 bins and 2 to 1024 bins.

Table 1: Comparison of FPGA slices against number of bins for DDC and PFT solutions

Tel: +44 1983 550330

If you have any comments to make about this article, or perhaps you require more information from the author, please email awinning@cmp-europe.com The resulting comments will be regularly posted on this site after the relevant article (CMP reserves the right to edit all comments before publication).