On the Use of Filter Banks for Parallel Digital Signal Processing

Phillip L. De Leon
New Mexico State University
Klipsch School of Electrical and Computer Engineering
Center for Space Telecommunications and Telemetering
Las Cruces, New Mexico 88003-8001
pdeleon@nmsu.edu, 505-646-1008

Abstract – **In parallel processing of digital signals, we require an algorithm which can be "parallelized" to take advantage of multiple processing units or a signal decomposition whereby each component in the signal decomposition can be processed in parallel. The filter bank is introduced as one way to provide a signal decomposition useful in parallel signal processing. In this paper, we review fundamentals of filter bank theory and provide a filter bank application suitable for parallel architectures.**

1 Introduction

In order to exploit parallel or Very-Long Instruction Word (VLIW) architectures for signal processing one needs an algorithm which can be suitably parallelized or partitioned to take advantage of multiple processing units [1]. If the algorithm does not have obvious tasks suitable for parallelization, algorithm transformation techniques such as program unfolding, retiming, index mapping, and/or look-ahead transformations can be applied for better processor utilization on concurrent systems [2]. Naturally, this approach relies on sophisticated compiler and operating system design and often places a significant burden on the applications developer. Another approach to parallel signal processing employs an orthogonal signal decomposition to partition the signal space. In this case, each component in the decomposition is processed in parallel. In this paper, we review basic filter bank theory and describe an FPGA-implementation of a polyphase, uniform-DFT filter bank. The implementation is to be used in a multiprocessor system suitable for high-bandwidth, complex parallel signal processing that is being designed at the Center for Space

This work was supported by NASA Grant NAG 5-1491.

Telecommunications and Telemetering. Finally, we describe one application which could exploit such an architecture.

2 Filterbanks for Parallel Signal Processing

One method for signal decomposition uses a set of bandpass filters (BPFs) denoted $\mathbf{f}_0$, ..., $\mathbf{f}_{M-1}$ in Figure 1 to uniformly partition the signal spectrum into subbands as in Figure 2. Once partitioned, the signals are then downsampled by a factor of D (due to their reduced bandwidth). Such a system of bandpass (analysis) filters and downsamplers is referred to as an analysis bank. Upon analysis, each of these subband signals (x_0, ..., x_{M-1}) may be processed (subband processing) independently of the others. After processing, the subband signals are synthesized into the fullband signal by first upsampling (zero insertion) to the original sampling rate followed by bandpass filtering ($\mathbf{g}_0$, ..., $\mathbf{g}_{M-1}$) to remove spectral images introduced from upsampling. Such a system of upsamplers and bandpass (synthesis) filters is referred to as a synthesis bank. The combination of analysis and synthesis banks is called a filter bank [3]. If the filter bank is designed properly, the parallel processing of the subband signals along with proper synthesis can often be made equivalent to fullband processing.

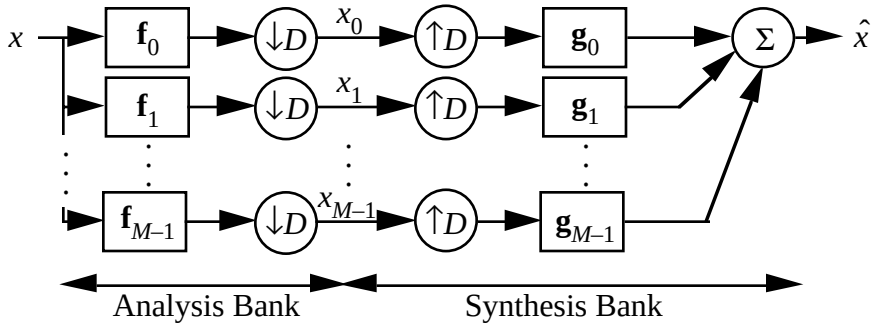

Figure 1: Filter bank (M subbands, M / D oversampled).

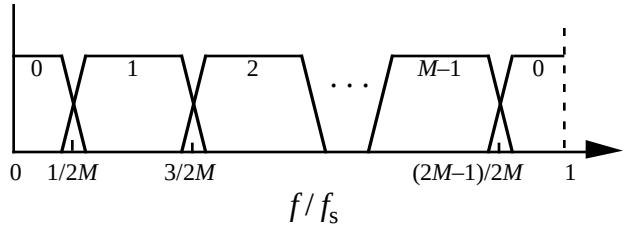

Figure 2: Uniform spectral partitioning (normalized sampling rate).

There are several advantages in using filter banks for parallel signal processing. First, each subband signal is independent of the others due to the near orthogonality of the decomposition and as such, the subband signals may be arbitrarily processed in parallel. In addition, the processing software of each signal component can often be made identical (as described in Section 6) and performed in lock-step. This can often be advantageous in VLIW architectures or in multiprocessor systems since many issues in interprocessor communications, synchronization, load balancing, and messaging can be bypassed. Second, due to the lower sampling rate associated with subband signals, the required performance of the subband processing unit is lowered. This would suggest that given adequate processor I/O, high bandwidth, complex signal processing could be accommodated. Third, the system is scaleable in that the number of subbands created in the signal decomposition can be matched to the available number of processing units or processors [4]. Finally, there is the option of bypassing the processing of certain subbands (as in subband coding) to reduce hardware/computational requirements. Naturally the cost in the filter bank approach for parallel signal processing is the latency associated in the analysis and synthesis which can, however, be controlled through judicious choice of analysis and synthesis filter length.

3 Polyphase, Uniform-DFT Filter Banks

In the analysis bank shown in Figure 1, much computation is wasted in the implementation since after filtering, only one out of every D samples is retained. Similar inefficiencies occur in the synthesis bank since only one out of every D samples is non-zero. A more efficient but equivalent implementation, called a polyphase, uniform DFT filter bank relies on a polyphase representation of $\mathbf{f}_0$ and $\mathbf{g}_0$ [lowpass filter (LPF) prototypes] and a discrete Fourier transform (DFT) and inverse DFT (IDFT) as shown in Figure 3 [6]. Note that in Figure 3, $I = M / D$ is referred to as the oversampling factor and will be discussed in the next section. The uniform DFT filter bank assumes that analysis and synthesis filters of Figure 1 are all generated from a simple frequency-shifting of the prototypes (normalized cutoff frequency of $1 / 2M$) described mathematically as [6]

$$\begin{aligned} \mathbf{f}_m(k) &= \mathbf{f}_0(k)e^{j2\pi km/M} \\ \mathbf{g}_m(k) &= \mathbf{g}_0(k)e^{j2\pi km/M} \end{aligned} \tag{1}$$

The polyphase representation of the prototype LPFs used in Figure 3 is given by

$$\begin{aligned} \mathbf{a}_m(k) &= \mathbf{f}_0(kD-m) \\ \mathbf{r}_m(k) &= \mathbf{g}_0(kD+m) \end{aligned} \tag{2}$$

where $m = 0, \ldots, M - 1$. With the polyphase representation, analysis and synthesis filtering occurs at the lower, subband sampling rate resulting in a lower processing speed [6]. In addition, the DFT and IDFT are implemented with a fast-Fourier transform (FFT). These combined effects result in a significant computational reduction.

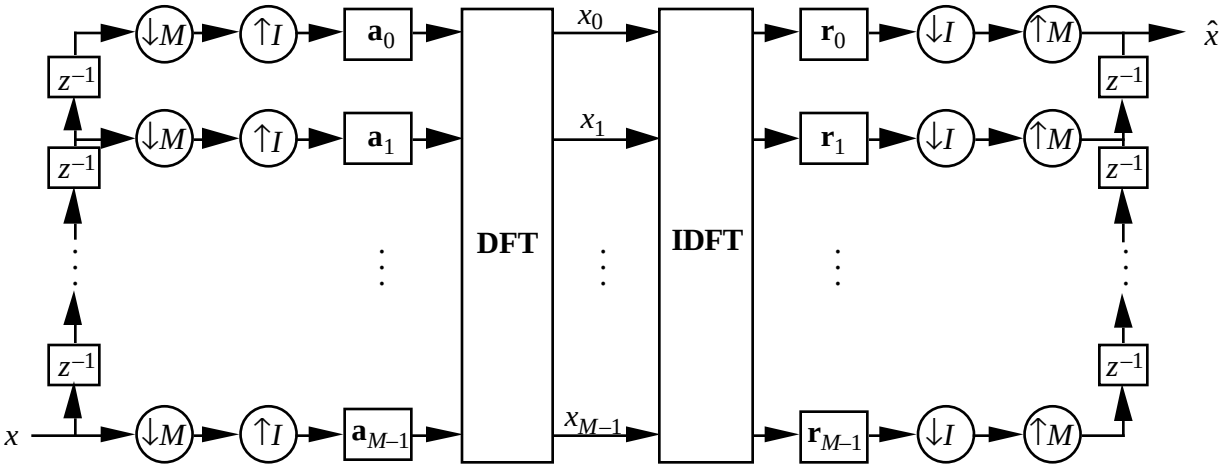

Figure 3: Polyphase, uniform DFT filter bank (*I*-times oversampled).

4 Subband Processing with Oversampled Subbands

In most of the literature, filter bank design is aimed at perfectly reconstructing x (to within a scale factor and delay) from the critically downsampled ($D = M$) subband signals x_0, ..., x_{M-1} (Figure 1). Critically downsampled refers to the fact that the number of fullband samples per second equals the total number of subband samples per second. In this case, one must carefully control the aliasing in the subband signals through proper design of the analysis and synthesis filters [3]. The fundamental problem in *processing* critically sampled subband signals is the large level aliasing which is present regardless of the analysis and synthesis filter designs due to the overlapping filters. With such a large amount of aliasing (which acts as a disruptive noise signal), many subband signal processing applications simply cannot be made equivalent to their fullband counterparts or they require a large additional overhead to partially compensate for the aliasing [5].

The alternate approach is to oversample ($D < M$) the subband signals to minimize the effect of subband aliasing. With properly designed analysis and synthesis filters, the oversampling factor I, can be made close (but not equal) to unity so that the total number of samples per second in the

subbands, although greater, is close to the fullband rate [6]. Even with oversampled subbands, many signal processing applications can have fewer total computations per sample as compared to the fullband case [7]. In any case, with minimal subband aliasing, parallel processing of the subbands can now be made equivalent to fullband processing (with delay).

In order to illustrate the fidelity of the reconstruction in an oversampled polyphase, uniform DFT filter bank, we measure the end-to-end impulse response of a sample filter bank. In the following, we assume a LPF prototype for both analysis and synthesis of length 29 (designed using MATLAB's remez function and shown in Figure 4a), $M = 4$ subbands, and downsampling by $D = 2$ [8]. This results in 2× oversampled subbands. The impulse response is given in Figure 4b. We note that the aliasing artifacts in the reconstruction are no greater than –44dB which nearly matches the quantization noise given 8 bit samples. In a perfect reconstruction filter bank (PRFB) these artifacts would be absent. Normally, when both analysis and synthesis filters are linear phase (as in our design) of length L, the filter bank delay is L. However, in the polyphase filter bank, if the prototype filter length is not an integer multiple of the number of subbands, we must zero pad the coefficient vector for the polyphase filters [see (2)]. Thus we note from Figure 4b the filter bank delay is 31 samples reflecting this zero padding. As mentioned earlier, the latency associated in subband processing is due to analysis and synthesis filter lengths. This latency can be decreased at the cost of increased subband aliasing since in general, FIR filter quality decreases with decreasing filter length. Decreased filter quality translates into increased aliasing due to more filter overlap in adjacent subbands and higher levels of signal energy folded back upon downsampling.

5 FPGA Implementation of Filter Bank

Efficient hardware implementation of the polyphase, uniform DFT filter bank is required in order to operate the parallel processors at very high sampling rates. We note from Figure 3 that the set of M polyphase filters can run in parallel provided we have a sufficient number of processing units. In addition both the FFT and IFFT are highly modular in the sense that the flow graphs share a common elementary (butterfly) computation. The butterfly computations can also be computed in parallel at each stage [9]. In addition, it is highly desirable that in any hardware implementation of the filter bank, the number of subbands can be scaled up or down.

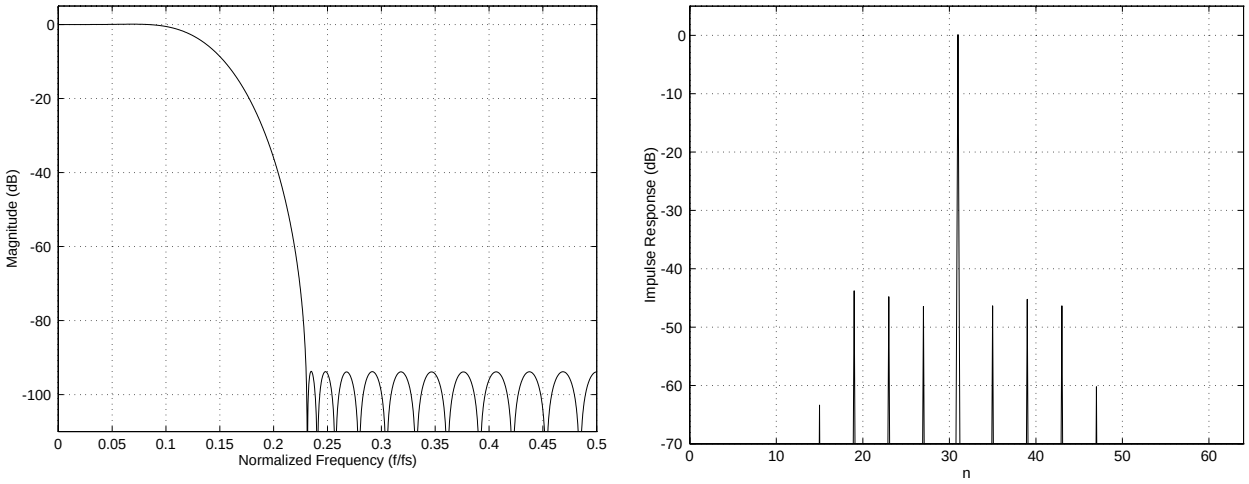

Figure 4: (a) Lowpass filter prototype. (b) Impulse response of 4-subband, 2× oversampled, polyphase, uniform DFT filter bank.

One implementation which exploits most of the architectural redundancy in the filter bank is the field-programmable gate array (FPGA). Multiple execution units which operate simultaneously can be constructed on the FPGA and through the use of distributed arithmetic, high-performance digital filtering can be accomplished [10], [11]. Furthermore, additional polyphase filters (and hence subbands) can be easily added (provided there is adequate area on the array) or removed by virtue of the programmability of the FPGA. Hence the parallel signal processing architecture can be easily scaled up or down. In addition to high-speed FIR filtering, the analyzer requires an FFT and the synthesizer an IFFT. Implementation of a pipelined FFT (butterfly computation nodes at each stage) on the FPGA provides an extremely efficient “FFT engine.” Again the pipelined FFT exploits the FPGA’s ability to run multiple units simultaneously. A 32-point FFT core with 16 bit word size can be constructed with 274 complex logic blocks (CLBs) and can operate at an input sample rate of 12MHz [12]. The IFFT is implemented using the same FFT engine but with input and output complex conjugated and scaled. The size of the FFT can be easily scaled on the array.

The 4-subband, 2× oversampled polyphase, uniform DFT filter bank given in Section 4 has been recently described in VHDL and implemented on a Xilinx FPGA at the Center for Space Telecommunications and Telemetering. This is the first step in constructing a parallel signal processing architecture utilizing subband decompositions. The length 16 polyphase filters (12 bit coefficient size) are derived from a length 29 prototype LPF (Figure 4a) and implemented with serial distributed arithmetic. The filter bank is capable of operating at an input sample rate of about

10MHz with 8 bit resolution. The set of polyphase filters require approximately 426 CLBs along with approximately 150 CLBs for the FFT. Additional overhead brings the total to approximately 800 CLBs (~20k gates) for the analysis bank with a similar CLB count for the synthesis bank. This implementation fits within a Xilinx XC4025E FPGA. In addition to processor array design to make use of the high-speed signal decomposition, work is underway to implement the polyphase filters with parallel distributed arithmetic which could increase the input sampling rate to beyond 50MHz with 8 bit resolution at the expense of increased CLBs.

6 Parallel Signal Processing Applications

There are several applications which could make use of signal decompositions to perform the signal processing in parallel [4]. In this section we discuss adaptive filters which are used in many applications such as adaptive equalization and acoustic echo cancellation. The adaptive adjustment of an FIR filter with a long impulse response (on the order of thousands of coefficients) presents a formidable computation problem [7]. One technique to overcome the computational problem is to replace the single, fullband adaptive filter, $\hat{\mathbf{h}}$ in Figure 5 with multiple, shorter-length, subband adaptive filters ($\hat{\mathbf{h}}_0,\cdots,\hat{\mathbf{h}}_{M-1}$) embedded in a filter bank structure as shown in Figure 6 [5].

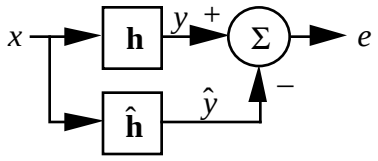

Figure 5: Fullband adaptive system identification.

It can be shown that the subband adaptive filter system can have fewer total computations while maintaining equal performance to the equivalent fullband system despite the added complexity of analysis and synthesis filtering [7]. More importantly, the subband adaptive filter system opens up the possibility of a multiprocessor implementation. In this case, a processor or processing unit can be dedicated to each subband adaptive filter whose adjustment can done independently of the others and in parallel while the analysis and synthesis filtering is performed on the FPGA. The update algorithm for each adaptive filter is identical and thus uses the same software operated in lockstep. In such a parallel arrangement, multiple, lower cost processors can be employed in the adaptive adjustment.

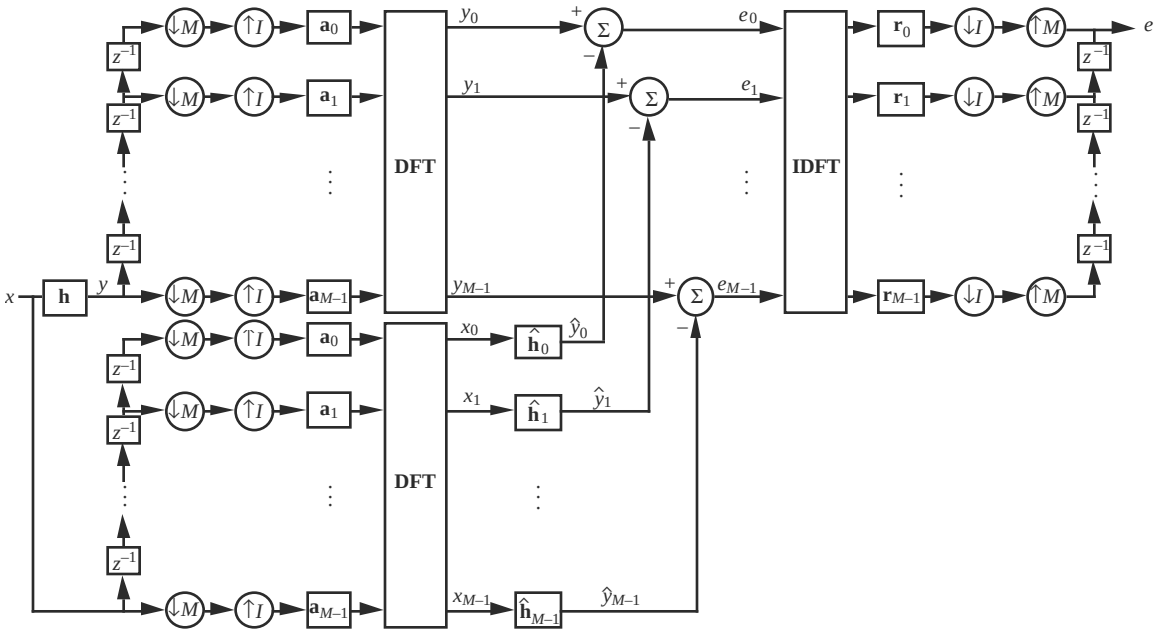

Figure 6: Subband adaptive filter system (system identification configuration).

7 Conclusions

In this paper we have reviewed basic filter bank theory emphasizing the oversampled, polyphase, uniform-DFT filter bank. This filter bank provides a signal decomposition that can be used in a parallel signal processing architecture. Such an architecture can be useful in high bandwidth, complex signal processing applications such as adaptive filtering. With oversampled subbands, processing of the subband signals can often be made equivalent to fullband processing and a simple design example is given. FPGA implementation of the filter bank example can take as few as 20k gates each for the analysis and synthesis sections and can operate at input sample rates of 10MHz or higher providing one avenue to high-bandwidth, complex parallel signal processing.

8 References

[1] P. Faraboschi, G. Desoli, J. Fisher, “The latest word in digital and media processing,” *IEEE Signal Processing Magazine*, vol. 15, no. 2, pp. 59-85, Mar. 1998.

[2] K. K. Parhi, “Algorithm transformation techniques for concurrent processors,” Proceedings of the IEEE, vol. 77, no. 12, pp. 1879-1895, Dec. 1989.

[3] Vaidyanathan, P. P., *Multirate Systems and Filter Banks*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[4] R. Sadr, P. P. Vaidyanathan, D. Raphaeli, S. Hinedi, "Parallel digital modem using multirate digital filter banks," *JPL Publication 94-20*, Aug. 1994.

[5] A. Gillore and M. Vetterli, "Adaptive filtering in subbands with critical sampling: analysis, experiments, and application to acoustic echo cancellation," *IEEE Trans. Signal Processing*, vol. 40, pp.1862-1875, Aug. 1992.

[6] L. Rabiner and R. Crochiere, *Multirate digital signal processing*, Englewood Cliffs, NJ: Prentice-Hall 1983.

[7] P. L. De León, "Optimization of the LMS Subband, Adaptive Filter System," *1997 International Conference on Signal Processing Applications & Technologies* (San Diego, CA.), 1997.

[8] P. De Leon and D. Morgan, "A simple condition for improving the overall response of linear phase, uniform-DFT filter banks," submitted to *IEEE Transactions on Signal Processing*, Jan. 1998.

[9] A. Oppenheim and R. Schafer, *Discrete-Time Signal Processing*, Englewood Cliffs, NJ: Prentice-Hall 1989.

[10] L. Mintzer, "FIR filters with the Xilinx FPGA," *FPGA'92 ACM/SIGDA Workshop on FPGAs*, 1992.

[11] B. Vonherzem, "Signal-processing at 250MHz using high-performance FPGA," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 6, no. 2, Jun. 1998.

[12] Xilinx Rev. 6 Q2 1998 AppLINX CD-ROM, Core Solutions DataBook, p. 2-65 - 2-68.