INTERNATIONAL CONFERENCE ON ACOUSTICS SPEECH AND SIGNAL PROCESSING - ICASSP, PHEONIX, ARIZONA, MARCH 15-19 1999.

HIGH-PERFORMANCE FPGA FILTERS USING SIGMA-DELTA MODULATION ENCODING

Chris Dick

Xilinx Inc.
2100 Logic Drive
San Jose, CA 95124, USA
chris.dick@xilinx.com

Fred Harris

College of Engineering
San Diego State University
San Diego
fred.harris@sdsu.edu

ABSTRACT

This paper investigates an architectural option for constructing high sample-rate narrow-band single rate and multi-rate filters using Xilinx field programmable gate array (FPGA) technology. Sigma-delta modulation encoding is applied to the input data in order to effect a reduction in the precision of the arithmetic units in the filter. This is done without compromising the signal integrity within the band of interest. The implementation provides a significant savings in device logic resources in comparison to other techniques that provide the same functionality. The sigma-delta pre-processor is described and its implementation using XC4000 FPGAs is reported. The architecture of the reduced precision filter is presented and its FPGA realization described.

1. INTRODUCTION

Narrow-band filters are utilized in many applications. For example, narrow-band communication receivers, multi-channel RF surveillance systems and for solving some spectrum management problems.

Data quantized by a uniform quantizer operating at the Nyquist rate is the standard solution to the problem of representing data within a specified dynamic range. Each additional bit of resolution in the quantizer provides an increase in dynamic range of approximately 6 dB. A signal with 60 dB of dynamic range requires 10 bits, while 16 bits can represent data with a dynamic range of 96 dB.

While the required dynamic range of a system fixes the number of bits required to represent the data, it also effects the expense of subsequent arithmetic operations, in particular multiplications. In any hardware implementation, and of course this includes FPGA based DSP processors, there are strong economic imperatives to minimize the number, and complexity, of the arithmetic components employed in the datapath. The proposal investigated in this paper is to employ noise-shaping to reduce the precision of the input data samples so that the complexity of the multiply-accumulate (MAC) units in the filter can be minimized. Of course, the pre-processing must not compromise the integrity of the signal in the band of interest. The net result is a reduction in the amount of FPGA logic resources required to realize the specified filter.

First, the overall system architecture employing noise-shaping techniques and a reduced complexity FIR filter is presented. The $\Sigma\Delta$ pre-processor is described in detail and its implementation using Xilinx XC4000 [1] devices is reported. The architecture of the reduced precision multiply-accumulate units used in the filter are also described. Next, a novel multi-rate architecture employing dual $\Sigma\Delta$ modulators is presented. The single-rate and multi-rate filters are compared to a conventional FPGA FIR filter implementation and shown to provide area savings of greater than 50% depending on the system specifications. Finally, some conclusions are drawn and several considerations regarding the applicability of the proposed $\Sigma\Delta$ modulator FIR are highlighted.

2. REDUCED COMPLEXITY FILTERS USING $\Sigma\Delta$ MODULATION TECHNIQUES

Consider the structure shown in Figure 1. Instead of applying the quantized data $x(n)$ from the analog-to-digital converter (A/D) directly to the filter, it will be pre-processed by a $\Sigma\Delta$ modulator.

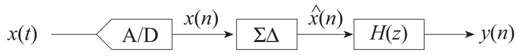


Figure 1: Reduced complexity FIR filtering employing a $\Sigma\Delta$ preprocessor.

The re-quantized input samples $\hat{x}(n)$ are now represented using fewer bits per sample, so permitting the subsequent filter $H(z)$ to employ reduced precision multipliers in the mechanization. The filter coefficients are still kept to a high precision.

The $\Sigma\Delta$ data re-quantizer is based on a single loop sigma-delta modulator [2]. In this configuration, the difference between the quantizer input and output sample is a measure of the quantization error which is fed back and subtracted from the next input sample. The error-feedback sigma-delta modulator operates on a highly oversampled input and uses the unit delay z^{-1} as a predictor. With this basic error feedback modulator only a small fraction of the bandwidth can be occupied by the required signal. In addition, the circuit only operates at baseband. A larger fraction of the Nyquist bandwidth can be made available and the modulator can be tuned if a more sophisticated error predictor is employed. This requires replacing the unit delay with a prediction filter $P(z)$. This generalized modulator is shown in Figure 2.

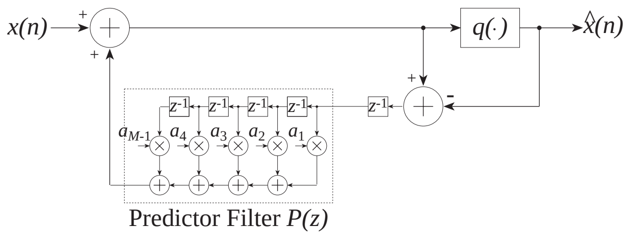


Figure 2: Tunable sigma-delta modulator using a linear filter in the feedback path.

The operation of the re-quantizer can be understood by considering the transform domain description of the circuit. This is expressed in Eq. (1) as

$$\hat{X}(z) = X(z) + Q(z)(1 - P(z)z^{-1}) \tag{1}$$

where $Q(z)$ is the z-transform of the equivalent noise source added by the quantizer $q(\cdot)$, $P(z)$ is the transfer function of the error predictor filter, and $X(z)$ and $\hat{X}(z)$ are the transforms of the system input and output respectively. $P(z)$ is designed to have unity gain and leading phase shift in the bandwidth of interest. Within the design bandwidth, the term $Q(z)(1 - P(z)z^{-1}) = 0$ and so $\hat{X}(z) = X(z)$. By designing $P(z)$ to be commensurate with the system passband specifications, the in-band spectrum of the re-quantizer output will ideally be the same as the corresponding spectral region of the input signal.

To illustrate the operation of the system consider the task of recovering a signal that occupies 10% of the available bandwidth and is centered at a normalized frequency of 0.3 Hz. The stopband requirement is to provide 60 dB of attenuation. Figure 3(a) shows the input test signal. It comprises an in-band component and two out-of-band tones that are to be rejected. Figure 3(b) is a frequency domain plot of the signal after it has been re-quantized to 4 bits of precision by a $\Sigma\Delta$ modulator employing an 8th order predictor in the feedback path. Notice that the 60 dB dynamic range requirement is supported in the bandwidth of interest, but that the out-of-band SNR has been compromised. This is of course acceptable, since the subsequent filtering operation will provide the necessary rejection. A 160-tap filter $H(z)$ satisfies the problem specifications. The frequency response of $H(z)$ using 12-bit filter coefficients is shown in Figure 3(c). Finally, $H(z)$ is applied to the reduced sample precision data stream $\hat{X}(z)$ to produce the spectrum shown in Figure 3(d). Observe that the desired tone has been recovered, the two out-of-band components have been rejected, and that the in-band dynamic range meets the 60 dB requirement.

3. ΣΔ MODULATOR FPGA IMPLEMENTATION

The most challenging aspect of implementing the data modulator is producing an efficient implementation for the pre-

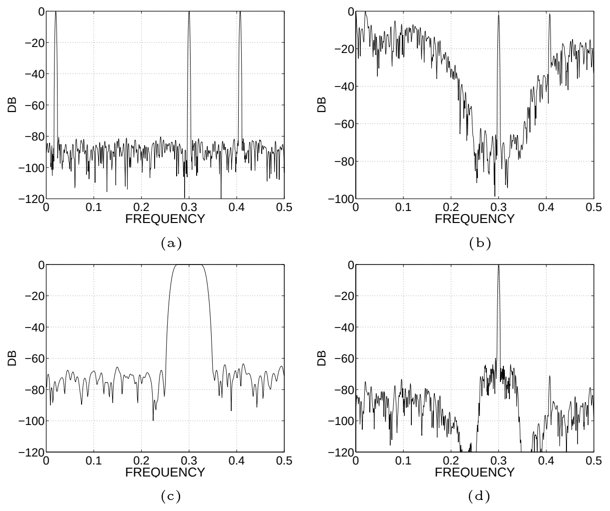
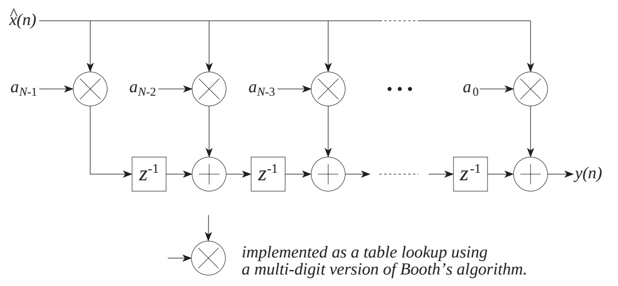


Figure 3: (a) Input signal - 12b samples. (b) Shaped input - 4b samples. (c) Filter response - 12b coefficients. (d) Filtered result.

diction filter $P(z)$. The desire to support high-sample rates, and the requirement of zero latency for $P(z)$, will preclude bit-serial methods from this problem. In addition, for the sake of area efficiency, parallel multipliers that exploit one time-invariant input operand (the filter coefficients) will be used, rather than general variable-variable multipliers. The *constant coefficient multiplier (CCM)* is based on a multi-bit inspection version of Booth's algorithm [3]. Partitioning the input variable into 4-bit nibbles is a convenient selection for the Xilinx XC4000 function generators (FG) [1]. Each FG has 4 inputs and can be used for combinatorial logic or as application RAM/ROM. Each configurable logic block (CLB) [1] in the XC4000 logic matrix comprises 2 FGs, and so can accommodate a 16×2 memory slice. Using the rule of thumb that each bit of filter coefficient precision contributes 5 dB to the sidelobe behavior, 12-bit precision is used for $P(z)$. 12-bit precision will also be employed for the input samples . There are 3 4-bit nibbles in each input sample. Concurrently, each nibble addresses independent 16×16 lookup tables (LUTs). The bit growth incorporated here allows for worst case filter coefficient scaling in $P(z)$. No pipeline stages are permitted in the multipliers because of $P(z)$'s location in the feedback path of the modulator. It is convenient to use the transposed FIR filter for constructing the predictor. This allows the adders and delay elements in this structure to occupy a single CLB slice. 64 CLBs are required to build the accumulate-delay path. The FPGA logic requirements for $P(z)$, using a 9-tap predictor, is $\Gamma(P(z)) = 9 \times 40 + 64 = 424$ CLBs. A small amount of additional logic is required to complete the entire $\Sigma\Delta$ modulator. The final CLB count is 450. The entire modulator comfortably operates with a 50 MHz clock. The critical path through this part of the design is related to the exclusion of pipelining in the multipliers.



$\hat{x}(n)$
a_{N-1}
a_{N-2}
a_{N-3}
a_0
z^{-1}
$y(n)$
implemented as a table lookup using
a multi-digit version of Booth's algorithm.

Figure 4: Area optimized FPGA FIR structure.

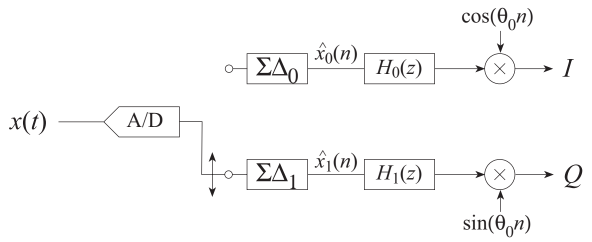


Figure 5: Dual half-rate $\Sigma\Delta$ modulators in m:1 complex decimator configuration.

4. REDUCED COMPLEXITY FIR MECHANIZATION

Now that the input signal is available as a reduced precision sample stream, filtering can be performed using area optimized hardware. For the reasons discussed above, 4-bit data samples are a convenient match for XC4000 devices. Figure 4 shows the structure of the reduced complexity FIR filter. The coded samples $\hat{x}(n)$ are presented to the address inputs of N coefficient LUTs. In accordance with the modulated data stream precision, each LUT stores the 16 possible scaled coefficient values for one tap. An N-tap filter requires N such elements. The outputs of the minimized multipliers are combined with an add-delay datapath to produce the final result. The logic requirement for the filter is $\Gamma(H(z)) = N\Gamma(MUL) + (N-1)\Gamma(ADD\_z^{-1})$ where $\Gamma(MUL)$ and $\Gamma(ADD\_z^{-1})$ are the FPGA area cost functions for a reduced complexity multiplier and an add-delay datapath component respectively. Each CCM occupies 40 CLBs, and 8 CLBs are required for an add-delay component. The total cost of a direct implementation of $H(z)$ is 7672 CLBs. The reduced complexity constant coefficient multipliers each consume 8 CLBs. Including the sigma-delta modulator the CLB count is 3002. So the data re-quantization approach consumes only 39% of the logic resources of a direct implementation.

5. $\Sigma\Delta$ DECIMATORS

The procedure for re-quantizing the source data can also be used effectively in an $m : 1$ decimation filter. An interesting problem is presented when high input sample rates ($\geq 100MHz$) must be supported in FPGA technology. High-performance multipliers are typically realized by incorporating pipelining in the design. This naturally introduces some latency in to the system. The location of the predictor filter $P(z)$ requires a zero-latency design.[1] Instead of re-quantizing, filtering and decimating, which would of course require a $\Sigma\Delta$ modulator running at the input sample rate, this sequence of operations must re-ordered to permit several slower modulators to be used in parallel. The process is performed by first decimating the signal, re-quantizing and then filtering. Now the $\Sigma\Delta$ modulators operate at the reduced output sample rate. This is depicted in Figure 5. To support arbitrary center frequencies, and any arbitrary, but integer, down-sampling factor m, the bandpass decimation filter must employ complex weights. The filter weights are of course just the bandpass modulated coefficients of a lowpass prototype filter designed to support the bandwidth of the target signal. Samples are collected from the A/D and alternated between the two modulators. Both modulators are identical and use the same predictor filter coefficients. The re-quantized samples are processed by an $m : 1$ complex polyphase filter to produce the decimated signal. Several design options are presented once the signal has been filtered and the sample rate lowered. Figure 5 illustrates one possibility. Now that the data rate has been reduced, the low rate signal is easily shifted to baseband with a simple, and area efficient, complex heterodyne. One multiplier and a single digital frequency synthesizer could be time shared to extract one or multiple channels.

It is interesting to investigate some of the changes that are required to support the $\Sigma\Delta$ decimator. What may not be immediately obvious is that the center frequency of the prediction filter must be designed to predict samples in the required spectral region in accordance with the output sample rate. For example, consider $m = 2$, and the required channel center frequency located at 0.1 Hz, normalized with respect to the input sample rate. The prediction filter must be designed with a center frequency located at 0.2 Hz. In addition, the *quality* of the prediction must be improved. With respect to the output sample rate, the predictors are required to operate over a wider fractional bandwidth. This implies more filter coefficients in $P(z)$. The increase in complexity of this component must of course be balanced against the savings that result in the reduced complexity filter stage to confirm that a net savings in logic requirements is produced. To more clearly demonstrate the approach, consider a 2:1 decimator, a channel center frequency at 0.2 Hz and a 60 dB dynamic range requirement.

Figure 6(a) shows the double sided spectrum of the input test signal. The input signal is commutated between $\Sigma\Delta_0$ and $\Sigma\Delta_1$ to produce the two low precision sequences $\hat{x}_0(n)$ and $\hat{x}_1(n)$. The respective spectrums of these two signals are shown in Figures 6(b) and 6(c). The complex decimation filter response is defined in Figure 6(d). After filtering, a complex sample stream supported at the low output sample rate is produced. This spectrum is shown in Figure 6(e). Observe that the out-of-band components in

[1]It is possible that the predictor could be modified to predict samples further ahead in the time series, but this potential modification will not be dealt with in the limited space available.

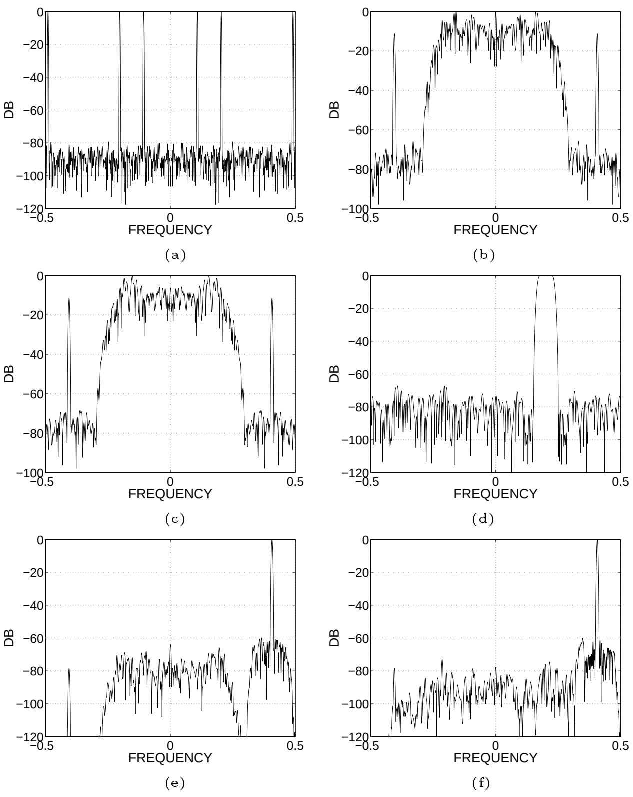


Figure 6: (a) Input signal. (b) Shaped data $\hat{x}_0(n)$. (c) Shaped data $\hat{x}_1(n)$. (d) Complex filter. (e) Recovered result. (f) Filtered signal - single modulator.

the test signal have been rejected by the specified amount and that the in-band data meets the 60 dB dynamic range requirement. For comparison, the signal spectrum resulting from applying the processing stages in the order, re-quantize, filter and decimate is shown in Figure 6(f). The interesting point to note is that while the dual $\Sigma\Delta$ modulator approach satisfies the system performance requirements, its out-of-band performance is not quite as good as the response depicted in Figure 6(f). The stopband performance of the dual modulator architecture has degraded by approximately 6 dB. This can be explained by noting that the shaping noise produced by each modulator is essentially statistically independent. Since there is no coupling between these two components prior to filtering, complete phase cancelation of the modulator noise cannot occur in the polyphase filter.

5.1. Discussion

To provide a frame of reference for the $\Sigma\Delta$ decimator, consider an implementation that does not pre-process the input data, but just applies it directly to a polyphase decimation filter. A complex filter processing real-valued data consumes double the FPGA resources of a filter with real weights. For $N = 160$, 15344 CLBs are required. This figure is based on a cost of 40 CLBs for each CCM and 8 CLBs for an add-delay component. Now consider the logic accounting for the dual modulator approach. The area cost $\Gamma(\widehat{\mathrm{FIR}})$ for this filter is

$$\Gamma(\widehat{\mathrm{FIR}}) = 2\Gamma(\Sigma\Delta) + \Gamma(\widehat{\mathrm{MUL}})) + \Gamma(\mathrm{ACC\_}z^{-1}) \qquad (2)$$

where $\Gamma(\Sigma\Delta)$ represents the logic requirements for one $\Sigma\Delta$ modulator, and $\Gamma(\widehat{\mathrm{MUL}})$ is the logic needed for a reduced precision multiplier. Using the filter specifications defined earlier, and 18-tap error prediction filters, $\Gamma(\widehat{\mathrm{FIR}}) = 2 \times 738 + 2 \times ((160 + 159) \times 8) = 6596$. Comparing the area requirements of the two options produces the ratio

$$\lambda = \frac{\Gamma(\widehat{\mathrm{FIR}})}{\Gamma\mathrm{FIR}} = 6596/15344 \approx 43\% \qquad (3)$$

So for this example, the re-quantization approach has produced a realization that is significantly more area efficient than a standard tapped-delay line implementation.

6. CENTER FREQUENCY TUNING

For both the single-rate and multi-rate $\Sigma\Delta$ based architectures, the center frequency is defined by the coefficients in the predictor filter and the coefficients in the primary filter. The constant coefficient multipliers can be constructed using the FPGA function generators configured as RAM elements. When the system center frequency is to be changed, the system control hardware would update all of the tables to reflect the new channel requirements. If only several channel locations are anticipated, separate configuration bit streams [1] could be stored, and the FPGA(s) re-configured as needed.

7. CONCLUSION

This paper has introduced a new alternative for constructing narrow-band filters using FPGA technology. The source data re-quantization approach is suitable for both single-rate and multi-rate processes. The proposed method arms the DSP/FPGA engineer with another tool that is useful for certain filtering requirements. For the examples considered here, logic savings in excess of 50% were demonstrated. As the frequency band of interest occupies a smaller fractional bandwidth, the order of the required filter increases. This growth tends to make the data re-quantization more attractive, as the cost of modulator consumes a decreasing proportion of the entire design.

8. REFERENCES

[1] Xilinx Inc., *The Programmable Logic Data Book*, 1998.

[2] J. C. Candy and G. C. Temes, “Oversampling Methods for A/D and D/A Conversion” in *Oversampling Delta-Sigma Data Converters - Theory Design and Simulation*, 1992, IEEE Press, New York.

[3] D. A. Patterson and J. L. Hennessy, *Computer Architecture: A Quantitative Approach*, Morgan Kaufmann Publishers Inc., California, 1990.