

Digital Audio Signal Processing Core

Maykel Post^{1,2}, Cornelis H. Slump¹, Louis Mulders²

¹University of Twente
Faculty of Electrical Engineering, Signals & Systems
PO box 217, 7500 AE Enschede, The Netherlands

²Axon Digital Design B.V.
Handelsweg 5, 5071 NT Udenhout, The Netherlands
E-mail: maykel@mad.scientist.com

Abstract— This paper presents a pilot project [9], which explored the possibility of a Digital Audio Signal Processing Core realization using an EPLD as a stand alone DSP-Filter-Engine. The implementation of a parametric equalizer in an EPLD had to clarify if present EPLDs are capable of containing complete audio processing applications. This paper presents the highlights of a full realization path of a parametric equalizer. Starting from off-line computer simulations performed with MATLAB till an EPLD implementation using an Altera FLEX 10K device. Error Spectrum Shaping (ESS) will be used to achieve the latest audio requirements of 24-bit resolution and a sample rate up to 96kHz. In many cases such as studio recording or during a live performance, continuous control of filter parameters is required. Taking advantage of the reconfigurability feature of the Embedded Array Blocks (EABs) in the Altera FLEX 10K device, it is possible to update the filter coefficients of the parametric equalizer in real-time, whereas the filter structure remains unchanged. This capability of the EPLD adds functionality and processing power to a minimum-chip DSP system controlled with an external processor/controller to calculate the filter coefficients. This Digital Audio Signal Processing Core likely can be applied in a broadcast audio mixer for post-production sound improvements.

I. INTRODUCTION

The adjustment and improvement of audio signals has been subjected to numerous research and has fascinated even more people. A variety of methods to process reproduced sounds have been developed over the years which can all be classified as fixed or variable equalizers. The last 30 years an explosion of variable equalizer developments could be observed, but it lasted till 1987 before the first digital equalizers entered the market. In 1987, Yamaha introduced the DEQ7 digital equalizer whereas Roland previewed a digital parametric equalizer.

High resolution Analog-to-Digital (A/D) and Digital-to-Analog (D/A) converters together with Digital Signal Processors (DSPs) are used as the transparent replace-

ment of many traditional analog audio processing functions. Parametric filters and shelving filters are used quite widely in the audio industry because of their ability to boost or to cut¹ specific frequency components of a signal. They are widely used in digital mixing desks, electronic musical instruments, and studios. All these products use the DSP as a signal processing core to realize a variety of filter functions [13].

Embedded Programmable Logic Devices (EPLDs) have become a competitive alternative for high performance digital signal processing applications, previously dominated by general purpose DSPs and Application Specific Integrated Circuit (ASIC) devices. EPLDs are standard, off-the-shelf user configurable Integrated Circuits (ICs) used to implement custom logic functions. At first, the EPLD was not designed for signal processing applications, but the progress of the last decade in EPLDs provide new options for DSP-based signal processing in EPLDs. In building a DSP system in an EPLD, the design can take advantage of parallel structures and arithmetic algorithms to minimize resources and to exceed the performance of single or multiple general-purpose DSP devices.

II. PARAMETRIC EQUALIZER FILTERS

Equalization (EQ) filters are used to modify the frequency response in a frequency band. A parametric equalizer is a variable equalizer offering control of all the "parameters" of the internal bandpass filter section. These *musical parameters*² are: gain, center frequency and Q-factor. This allows the user to not only control the amplitude of each band, but also to shift the center frequency and widen or narrow the pertinent area. Q is a measurement of how much the equalizer band affects a range of frequencies.

A widely used technique in the design of IIR digital filters with a prescribed magnitude response is the bilinear

¹Original terms used by Art Davis to signify direction of equalization. Equalize means to make bigger and attenuate means to make smaller. Replaced today by *boost/cut* terminology.

²Electronic engineers are used to think in terms of bandwidth, passband ripple, slope etc., whereas the musicians and recording engineers are more comfortable thinking in terms of the *musical* filter parameters: gain, center frequency and Q-factor. From the viewpoint of the user, the *musical* filter parameters are used to control the parametric equalizer.

transformation method. The overall design method can be summarized as follows [3]: starting with given magnitude response specifications for the digital filter, the specifications are transformed by the appropriate *prewarping transformation* into the specifications of an equivalent analog filter $H(s)$. Using an analog filter design technique, the equivalent analog filter $H(s)$ is designed. Using the *bilinear transformation* the analog filter is mapped back into the desired digital filter $H(z)$. Fig. 1 shows the frequency response of a typically second-order parametric equalizer. The filter specifications are:

- G_0 , reference gain (typically unity for cascadable filters)
- G , filter gain at ω_0
- ω_0 , center frequency of the boost or cut
- $\Delta\omega$, bandwidth at level G_B

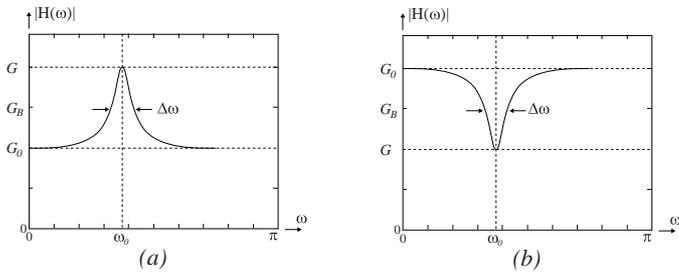


Fig. 1. Frequency response of a parametric equalizer: (a) boost (b) cut.

The definition of the bandwidth $\Delta\omega$ of the boost or cut of a parametric equalizer is arbitrary. As discussed by [1], [8], there is considerable variation in the literature in the definition of bandwidth $\Delta\omega$ and bandwidth gain G_B . The requirements the parametric equalizer has to satisfy are:

- Gain: 18 dB to -18 dB
- Center frequency: 20 Hz to 20 kHz
- Q-factor: 0,25 to 10
- Word length: 24 bit
- Sample rate: 96 kHz

The filter design problem is to realize the transfer function in terms of the specification parameters: $\{\omega_0, \Delta\omega, G_B, G, G_0\}$. A second-order analog equalizer with gain G_0 at frequency zero has the transfer function

$$H(s) = \frac{G_0 s^2 + Bs + G_0 \Omega_0^2}{s^2 + As + \Omega_0^2} \quad (1)$$

After the filter parameters $\{\omega_0, \Delta\omega, G_B, G, G_0\}$ are mapped into the analog domain, using the prewarping transformation given by

$$\Omega_0 = \tan\left(\frac{\omega_0}{2}\right), \quad A = \sqrt{\frac{G_B^2 - G_0^2}{G^2 - G_B^2}} (1 + \Omega_0^2) \tan\left(\frac{\Delta\omega}{2}\right),$$

$$B = GA \quad (2)$$

the analog filter is mapped into the digital domain by the bilinear transformation

$$s = \frac{1 - z^{-1}}{1 + z^{-1}} \quad (3)$$

resulting in

$$H(z) = \frac{\left(\frac{G_0 + G_B}{1 + \beta}\right) - 2\left(\frac{G_0 \cos \omega_0}{1 + \beta}\right) z^{-1} + \left(\frac{G_0 - G_B}{1 + \beta}\right) z^{-2}}{1 - 2\left(\frac{\cos \omega_0}{1 + \beta}\right) z^{-1} + \left(\frac{1 - \beta}{1 + \beta}\right) z^{-2}} \quad (4)$$

The parameter β is given by

$$\beta = \sqrt{\frac{G_B^2 - G_0^2}{G^2 - G_B^2}} \tan\left(\frac{\Delta\omega}{2}\right) \quad (5)$$

The transfer function $H(z)$ for a second-order IIR filter topology is given by

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \quad (6)$$

Using Eq. (4) and Eq. (5) the filter coefficients $\{b_0, b_1, b_2, a_1, a_2\}$ of Eq. (6) can be expressed in terms of $\{\omega_0, \Delta\omega, G_B, G, G_0\}$. The terms G_0 and G_B are set once and depends on the application form. For cascadable filters the gain at frequency zero and Nyquist is usually set to $G_0 = 1$, whereas the bandwidth G_B can be defined as described in [1], [8]. The musical parameters, gain g , center frequency f_0 , and selectivity Q , determine the terms: G , ω_0 and $\Delta\omega$, and vary in time if another filter function is required.

III. FILTER STRUCTURES

Once the transfer function of the parametric equalizer is obtained the design of this transfer function is the next step. Of course, a transfer function specifies the filter coefficients, but does not specify how a filter is designed. Choosing the best filter topology is one of the most difficult jobs facing the digital audio engineer. Issues of controlling noise, scaling signal amplitude, finite-word-length effects, and preventing overflow are all difficult to solve. Some filter structures which have been examined are [9]:

- Direct form I (for audio use)
- Direct Form II (not for audio use)
- Transposed Direct Form I (not for audio use)
- Transposed Direct Form II (for audio use)
- Lattice-Ladder (not for audio use)

In order for a digital filter structure to be suitable for high-quality audio, it must not degrade system performance in such a way that the system specifications are violated. This means that the effects of the filter must be transparent to the user, except for the intended changes in frequency and phase response. The digital filter is a component of the total audio system, therefore it must have specifications that exceeds those of the host system. Traditionally the direct form II topology has been recommended in comparison to the direct form I, as it has superior noise performance when both are implemented with truncation after each multiplication. Otherwise the direct form I is better [11].

Lattice topologies have good noise performance [6], [10]. The noise amplitude does not vary as a function of the pole zero positions. Instead it is a constant, depending only on the filter order. This property is certainly not true for the direct form topologies, where the noise amplitude depends

on the pole zero positions. Two drawbacks of the lattice-ladder structure are its sensitivity to signal overflow, and its audible low-level zero-input limit-cycle tones [9]. The total implementation cost of the lattice-ladder structure is much higher than a direct-form structure for the same order filter. This is the main reason why this filter has not found much applications.

Topologies that require additional input scaling to prevent overflow at internal nodes should not be used for audio purposes. Scaling down the filter input to prevent overflow generally reduces the SNR by an amount proportional to the input scaling. Both the direct form II and the transposed direct form I topologies require additional input scaling to prevent overflow at internal nodes and so are not recommended for audio purposes [2]. We can conclude that the direct form I is the best of the direct and transposed forms for audio use. The direct form I is also superior to the lattice topology with regard to, topology simplicity, small-scale zero-input limit cycle tones and overflow properties. Because there is no need to scale the input signal, no reduction of the SNR occurs. We end this discussion about filter structures by concluding that the direct form I is the best and most elegant digital filter topology for audio purposes. This conclusion holds, only if we control its truncation noise.

IV. NONLINEAR ASPECTS OF DIGITAL FILTER DESIGNS

For any given difference equation, the designs are equivalent in that they represent the same system and produce the same output for any given input provided that the internal computations are performed with infinite-precision arithmetic. However, the various structures are not equivalent when the internal computations are performed with finite-precision arithmetic. Finite-word-length effects of digital filters have been the subject of much research in recent years [4], [7], [12]. The following nonlinear aspects of digital filters have to be considered:

- Coefficient quantization
- Signal truncation errors
- Forced overflow oscillations

The first two types which are also known as quantization effects should be observed in any design. The first is due to *coefficient quantization*, where the term *coefficient* refers to the fixed digital filter coefficients. The result of coefficient quantization is that the actual transfer function differs from the desired function. Once the quantization has been done, this error is a fixed, well-determined quantity. The second type of quantization is due to *signal truncation*³. The phenomenon of signal truncation can be analyzed by suitable noise modeling and by the use of linear system theory. A third type of nonlinearity can be introduced due to *forced overflow oscillations*⁴ which comes into effect if the mag-

³The topic is collectively referred to as *roundoff noise*. Dat-torro [2] substituted the term *truncation* since it is this and not rounding which will be performed in hardware implementations. For this good reason the term truncation is used.

⁴The topic of *forced overflow oscillations* should not be confused with the more widely researched topics of overflow os-

truncation of input signals of a digital filter is not considered and accumulation overflow occurs.

A. Coefficient quantization

While input and output word length are fixed by a specific format, filter coefficient lengths as well as partial results are determined by the implementation form. In a DSP implementation the word length of the filter coefficients is limited because the numbers are stored in finite-length registers. Using an EPLD implementation, register lengths can be optimized with respect to the required word length of the coefficients. If coefficient quantization is applied, the coefficient values change and another filter approximation is obtained. The poles and zeros of the system function will in general be different from the desired poles and zeros. Consequently, we obtain a filter having a frequency response that is different from the frequency response of the filter with unquantized coefficients. If the quantization step is coarse, the filter may actually fail to meet the desired specifications.

A drawback of the direct and transposed forms is the distribution of poles that can be implemented by these topologies. The sensitivity of filter coefficient quantization can greatly be reduced using the low-sensitivity structures mentioned in [9], [12], however, the price that we pay for this change in distribution of the pole positions is an increase in structure complexity. The Gold-Rader structure requires four multiplications per output point, whereas the direct form realization requires only two multiplications per output point. The Gold-Rader, Kingsbury and Zölzer topologies [13] contain internal nodes which require scaling of the input signal. Signal scaling reduces the SNR, which is not recommended for practical audio purposes. Due to their complexity, these structures are often more interesting from the theoretical point of view than real alternatives for practical implementation.

For a minimum logic-cell EPLD implementation, it is much more economical to reduce sensitivity by applying more pole positions to the structure. In other words, if the filter design needs a higher pole density in a certain region then simply more bits should be allocated to the coefficients. It was observed that most filters used in high-fidelity audio can be implemented by using 24 bit coefficients [9].

V. ERROR SPECTRUM SHAPING

Quantization operations arise in multipliers and adders of digital filter structures and are unavoidable. When the input of a filter is M bit (sign included), the internal word length produced by the multipliers and accumulators will be *at least* $(2M - 1)$ bit. Usually the output is M bit, so that regardless of the accumulated word length, the filter output must be truncated to M -bit signal data. Errors arise from this truncation process are trapped into the feedback loop of a recursive filter structure where they can be

cillations and limit cycles. These are two types of low-level autonomous oscillations possible in digital filters while forced overflow oscillations deal with large signals.

amplified by the poles of the digital filter. For ideal linear filters, the transfer function is supposed to be independent of the input signal level. By truncating the output of the accumulator a nonlinearity is introduced which results in an *input-signal-dependent* filter transfer function. This effect becomes noticeable when low-level signals are input, and is even worse at low input frequencies [2].

Error Spectrum Shaping (ESS) [2], also known as **Error Feedback (EF)** [4], [5], noise shaping, or residue feedback, is a method that can be used to reduce errors which arise from the truncation operations in digital filter realizations. This technique involves the generation of an error signal and the application of local feedback for the purpose of manipulating the output truncation noise. The truncation error is stored and fed back through a separate filter to compensate the next truncation, resulting in a reduced overall error if the coefficients are appropriately chosen. This technique entails additional hardware which increases the number of adders and/or multipliers in the structure.

A. Noise analysis of the direct form I structure

The direct form I structure with quantizer is depicted in Fig. 2. The N -bit quantity $y(n)$ is the ideal fixed-point output of the digital filter, while the M -bit quantity $y_{trunc}(n)$ is the truncated output.

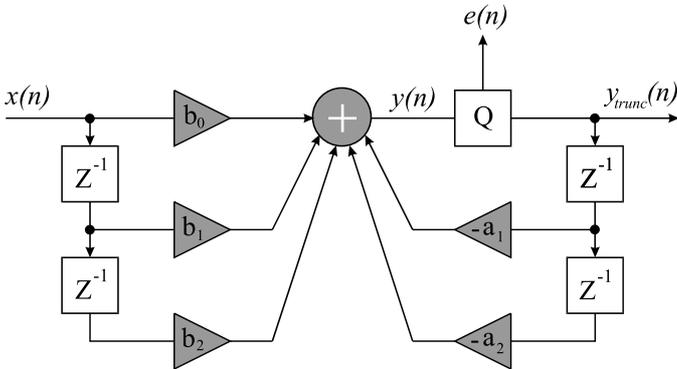


Fig. 2. Direct form I structure with access to the truncation error $e(n)$.

The frequency-domain expression of the direct form I structure, using the truncated output in the feedback loop, is given by

$$Y_{trunc}(z) = \frac{\sum b_i z^{-i}}{1 + \sum a_i z^{-i}} X(z) - \frac{1}{1 + \sum a_i z^{-i}} E(z) \quad (7)$$

From Eq. (7), we see that the output spectrum of the filter is a combination of the input spectrum $X(z)$ multiplied by the filter transfer function, and an *error function* multiplying the truncation error spectrum $E(z)$. It is clear that the truncation error spectrum is amplified by the poles of the filter transfer function⁵. Any improvements made to minimize the effects of truncation error have to be done on the last term of Eq. (7).

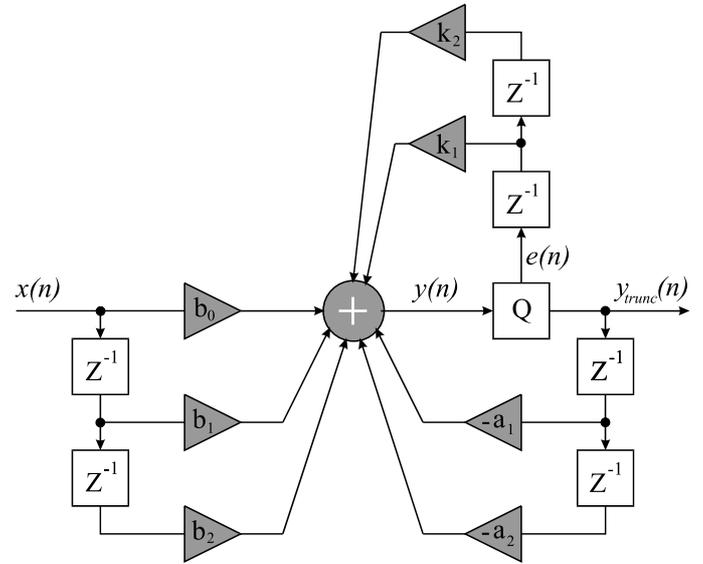


Fig. 3. Direct form I structure with second-order error reduction. The feedback coefficients are, k_1 and k_2 .

B. Noise reduction feedback

An error feedback scheme is shown in Fig. 3. The frequency-domain expression of the direct form I structure, using the truncated output in the feedback loop, and applying a second-order error feedback, characterized by coefficients k_1 and k_2 , is given by

$$Y_{trunc}(z) = \frac{\sum b_i z^{-i}}{1 + \sum a_i z^{-i}} X(z) - \frac{1 - k_1 z^{-1} - k_2 z^{-2}}{1 + \sum a_i z^{-i}} E(z) \quad (8)$$

Placing the error feedback zeros on the unit circle close to the poles provides a near-optimum solution when the poles are close to the unit circle. This results in a nearly flat noise spectrum at the filter output. First-order solutions have one parameter which set a fixed real zero at $z = \pm 1$, suitable for narrow-band, low-pass, or high-pass filters when only moderate noise reduction is required. The second-order solution contains 2 free parameters which control the locations of the complex-conjugate zeros, offering more efficient noise reducing capabilities. These are best suited for high quality narrow-band low-pass, or high-pass-filters.

Applying the second-order feedback circuit to the parametric equalizer, and with the knowledge that the equalizer will operate in the low-frequency region, with respect to the sample rate of 96 kHz, we have the opportunity to implement two zeros at $z = 1$. This results in error coefficients, $k_1 = 2$ and $k_2 = -1$.

C. Noise cancellation feedback

Another ESS scheme, known as the optimal ESS filter, is optimal over the entire band, but more intensive than the error feedback schemes discussed so far. If we set $k_1 = -a_1$ and $k_2 = -a_2$ in Fig 3, then the frequency-domain

⁵ $E(z)$ is not influenced by the feed forward paths (FIR-structure) of the direct form I structure.

expression of the Direct Form I with second-order feedback becomes

$$Y_{trunc}(z) = \frac{\sum b_i z^{-i}}{1 + \sum a_i z^{-i}} X(z) - E(z) \quad (9)$$

The optimal ESS structures introduces zeros in the noise spectrum at exactly the same location as the poles and so eliminates the noise gain caused by the poles. This is the best we can achieve⁶. Optimal ESS is exactly equivalent to extending the word length of the internal variables of the filter. The length of the stored coefficients remains unchanged. Since optimal ESS uses the same coefficients as the recursive part of the filter transfer function, no additional computation is required to compute the ESS filter coefficients.

The purpose of truncation error feedback is to make the actual frequency response less dependent on absolute input signal level. As we have seen ESS can significantly reduce output noise levels in narrow band digital filters with only a slight increase in hardware complexity. Error feedback effects only the transfer function of the error signal, while the transfer function of the filter itself remains unchanged. ESS cannot have any effect on the coefficient sensitivity properties of a filter structure, neither can it enhance the overflow properties of the filter implementation.

For EPLD purposes, suboptimal ESS can be achieved using power-of-two coefficients resulting in simple shift and add techniques. Optimal ESS is the best we can get but needs two additional multipliers in the error feedback circuit which leads to a cell-consuming solution.

VI. IMPLEMENTATION

Implementation of a structure is a trade off between parallel and serial solutions. Parallel structures require the most cells, but can do the processing in a minimum number of clock cycles. With a fully parallel design the sample speed can match the system clock rate. A serial solution requires a minimum of logic cells but needs the most clock cycles to do the job.

The multipliers of the parametric equalizer are implemented as bit-serial ones. On the other hand all the multipliers operate in parallel otherwise the overall filter operation would exceed the maximum amount of available clock cycles.

A. The bit-serial multiplier

The multiplier used in the filter structure is a bit-serial one. Data input of 24 bit is multiplied by a filter coefficient of 27 bit. The resulting word length is 51 bits. The multiplier is depicted in Fig. 4.

The multiplier forms partial products by multiplying the input data $mul_in[23..0]$ by 1 bit of the coefficient co

⁶For an optimal ESS structure the truncation error spectrum $E(z)$ is not amplified by the poles of the transfer function $H(z)$. The error function is decoupled from the filter transfer function. The output will be in error by at most 1 LSB in magnitude. The truncation process will add only 6 dB to the noise floor in the computation of $y_{trunc}(n)$.

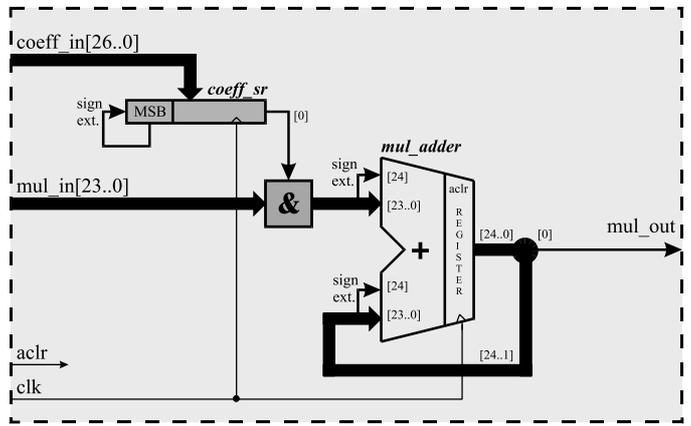


Fig. 4. Blockdiagram of the 24 × 27 bit-serial multiplier.

$eff_in[26..0]$ at a time in an AND operation. This technique then adds the partial products into an accumulator mul_adder that shifts the accumulator's feedback 1-bit position, to the right, to perform a divide by two each clock cycle, thus compensating for the bit-weighting of the ingressing partial product. Each clock cycle the output $mul_out[0]$ of the accumulator represents a bit in order to construct the result. This process requires $n + p + 1$ clock cycles for data words of n -bits and coefficient length of p -bits (and one pipeline clock cycle). The last partial product, presented at $mul_out[27..0]$ after $p + 1$ clock cycles, is properly out-shifted by a sign extended⁷ coefficient. The timingsdiagram of the serial multiplier is shown in Fig. 5.

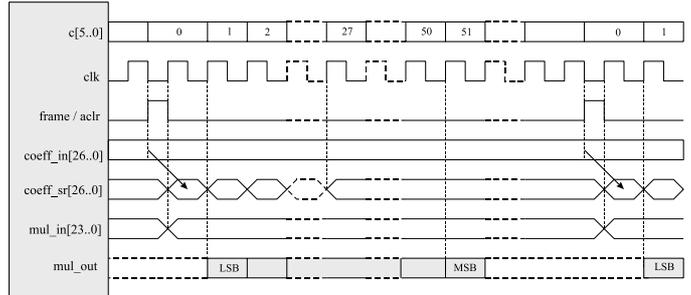


Fig. 5. Timingsdiagram of the 24 × 27 bit-serial multiplier.

On every rising edge of the frame pulse a new coefficient is written into the $coeff_sr[]$ register, where the FSYNC-line takes care of loading the proper coefficient (left or right channel). This design-feature of the multiplier enables a **coefficient-update on sample basis**. For an Altera FLEX 10K implementation, the bit-serial multiplier consumes about 90 logic cells and the maximum clock frequency of this circuit is 48.78 MHz.

B. The bit-serial adder

The Direct Form I with noise reduction contains a 7-input bit-serial adder. The blockdiagram of the 7-input bit-serial adder is depicted in Fig. 6. Every clock cycle seven

⁷Sign extended: Extending the word length of a coefficient at the MSB-side with b bits. These b bits are a copy of the MSB, which represents the proper sign of the coefficient.

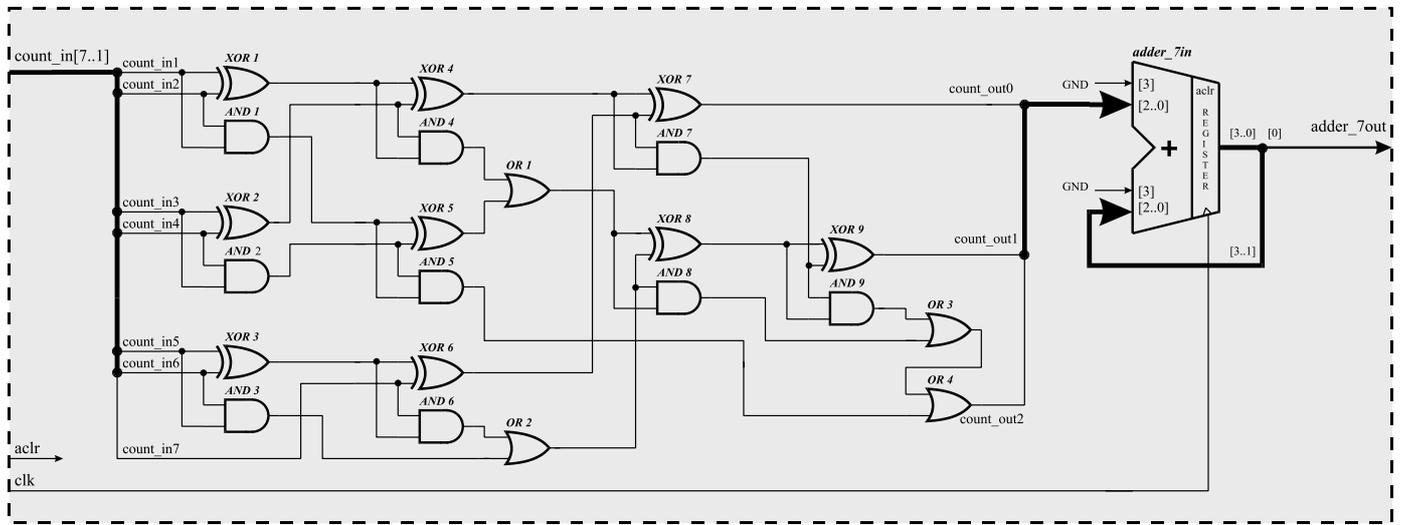


Fig. 6. Blockdiagram of the 7-input bit-serial adder.

bits are added, resulting in a 3-bit partial word. The partial words are added into an accumulator, *adder_7in*, that shifts the accumulators feedback 1-bit position, to the right, to perform a divide by two each clock cycle. This is the same principle as used in the serial multiplier. Each clock cycle the output *adder_7out*[0] of the accumulator represent a bit in order to construct the result. This process requires $s + 1$ clock cycles, where $s = n + p$. The sign extensions of the *adder_7in* are connected to ground because the partial words present at *count_out*[*i*] are **always** positive signed.

The process to construct the result at the output of a multiplier requires $n + p + 1$ clock cycles. The process to add all the results of the multipliers in the bit-serial adder requires also $n + p + 1$ clock cycles. In a worst case situation a complete filter operation would require $2 \times (n + p + 1)$ clock cycles. By optimizing these two operations the overall processing time of the multiplier and adder operation is reduced to $n + p + 2$ clock cycles. The timingsdiagram of the 7-input adder is shown in Fig. 7. The bit-serial adder consumes only 17 logic cells and can handle a system frequency up to 125 MHz.

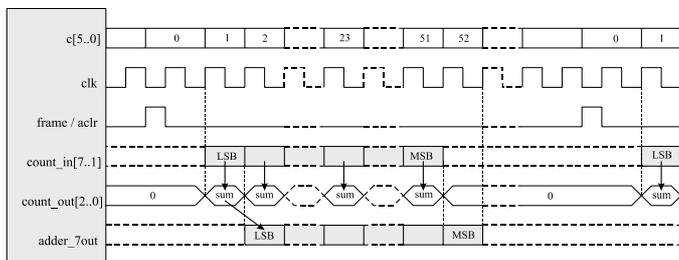


Fig. 7. Timingsdiagram of the 7-input bit-serial adder.

C. Direct form I with noise reduction

The structure of Fig. 8 includes a feedback circuit with a second-order noise reduction topology. The *equalizer_out*[51..0] is a 52-bit register which includes the forced overflow correction. Bit [47..24] of this register contains the

24-bit result of the filter-operation, whereas bit [51..47] are used to detect and correct forced overflow.

Bit [23..0] represents the truncation error, which is fed back through the error circuit. All the multipliers in this topology uses one additional pipeline clock-delay, as seen in Fig. 8. This clock delay is necessary to accomplish a proper bit adjustment when the results of the five multipliers and the result of the error circuit are summed into the 7-input adder. The second-order error feedback circuit uses two ESS coefficients with power-of-two values which will be explained in the next subsections. As already mentioned, two zeros at $z = 1$ will be implemented, which results in the error coefficients, $k_1 = 2$ and $k_2 = -1$.

Embedding both left and right channel into the same filter structure reduces the amount of logic cells by a factor 2, where the filter coefficients of the left and right channel are properly loaded (every frame) into the *coeff_sr* registers of the multipliers.

C.1 Error multiplier k_1

The error multiplier coefficient $k_1 = 2$ is realized using a simple shift technique. The blockdiagram of the *error multiplier* k_1 is depicted in Fig. 9. By shifting the data of the *error_shift_k1* register one bit to the left a multiply factor of 2 is created. The same effect can be realized by reading the 24-bit positive signed fractional part of *delay_reg9*[23..0] into the 25-bit *error_shift_k1* register on position [24..1] where the first bit position is filled with a zero. The timingsdiagram of the *error multiplier* k_1 is shown in Fig. 10.

The D-flip-flop in this multiplier is used as a time delay to adjust the LSB of the result with the other multiplier results for a proper addition into the 7-input adder (it has the same function as the pipeline clock-delay of the multipliers). After multiplying the positive signed fractional part with 2 the result at *error_out_k1* is **always** a 26 bit positive signed fraction part (24 error bits one position left shifted and an additional sign bit). This result has to be sign extended to 52 bits for a proper addition into the 7-

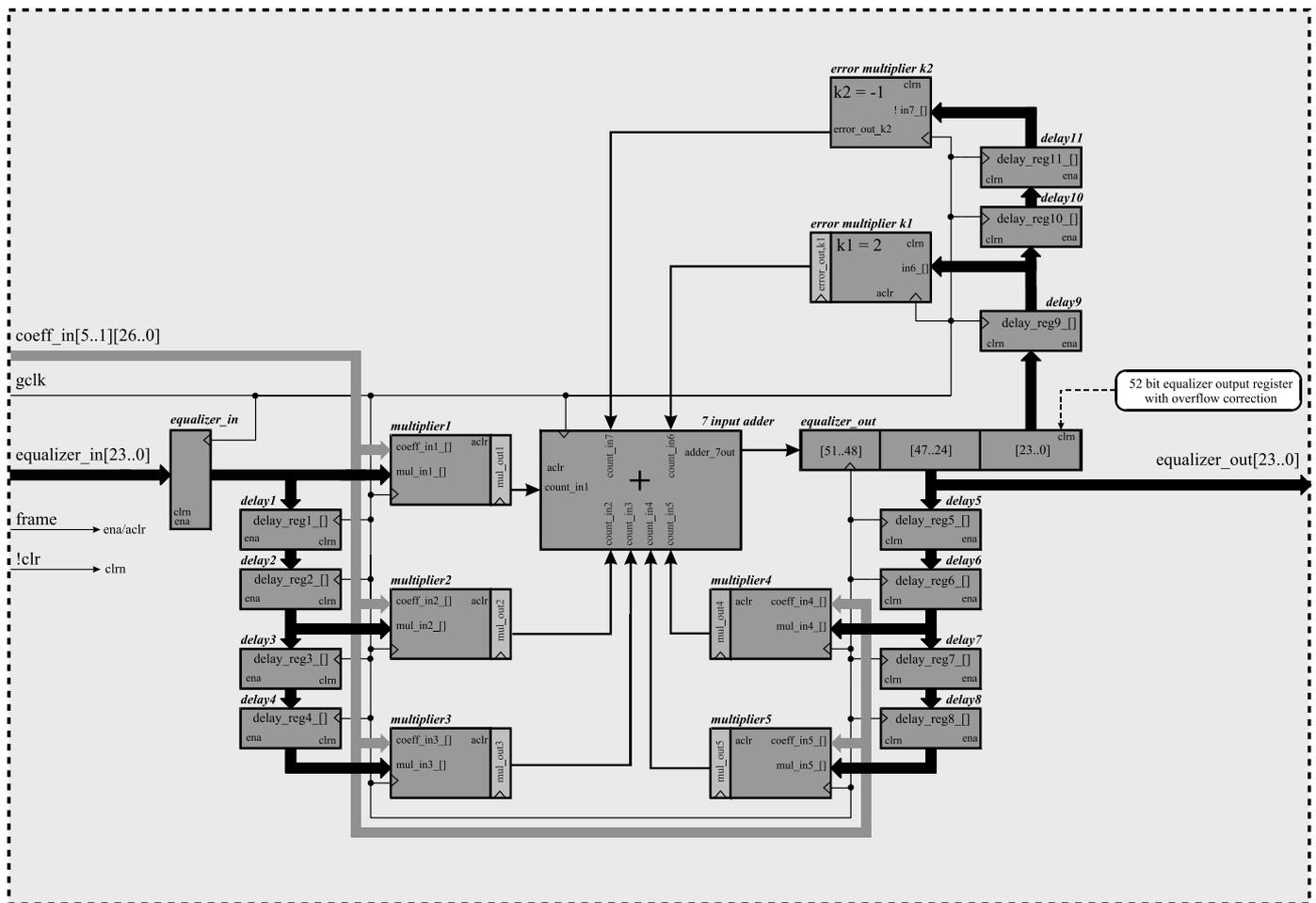


Fig. 8. Blockdiagram of the stereo direct form I structure with second-order noise reduction.

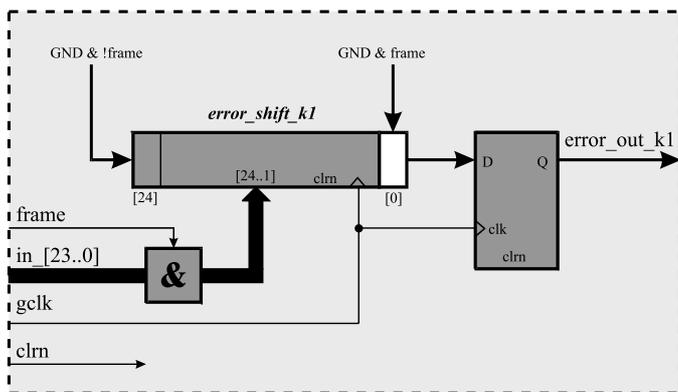


Fig. 9. Blockdiagram of the error multiplier k_1 .

in adder. In the end, $error_out_k_1$ represents a serial bit stream where bit [51..25] and [0] are filled with zeros and bit [24..1] representing the truncation error.

C.2 Error multiplier k_2

In addition, the error multiplier coefficient $k_2 = -1$ can also be realized using simple shift and add techniques. Negate (i.e., complement) a two's complement number is simply accomplished by replacing each 1 with a 0 (and each 0 with a 1), including the sign bit, and then adding the

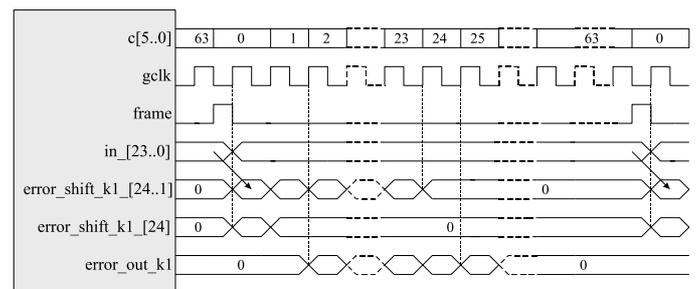


Fig. 10. Timingsdiagram of the error multiplier k_1 .

LSB. The blockdiagram and timingsdiagram of the *error multiplier k_2* are depicted in Fig. 11 and Fig. 12.

The data read from $delay_reg11[23..0]$ is inverted and written into the register $error_shift_k_2$. Then it is serially clocked into the *complement adder* where the LSB is added. The clock cycle delay introduced by the registered output of the *complement adder* is the reason why the outputs of the other multipliers needs an extra pipeline clock-delay. After multiplying the positive signed fractional part with -1 the result is **always** a 25 bit negative signed fractional part. This result has to be sign extended to 52 bits for a proper addition into the *7 input adder*. This time, bit [51..24] of the serial bit stream at $error_out_k_2$ will be filled with ones,

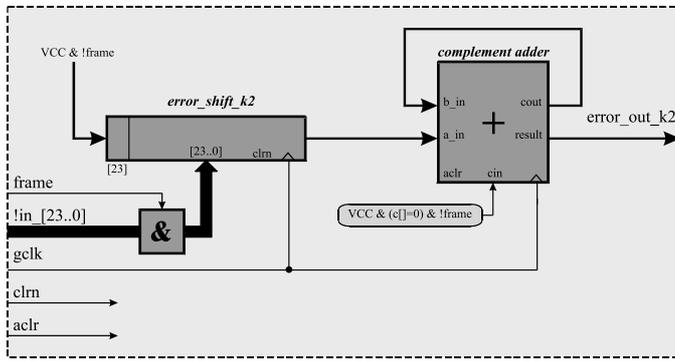


Fig. 11. Blockdiagram of the error multiplier k_2 .

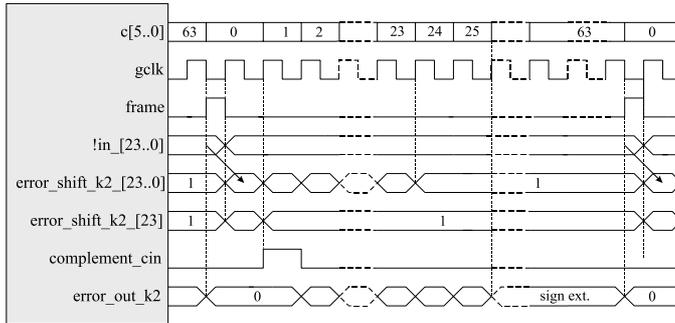


Fig. 12. Timingdiagram of the error multiplier k_2 .

whereas bit [23.0] represents the negated truncation error.

D. LUT-based SDA direct form I with noise reduction

It is observed that the main part of logic cells is consumed by the multipliers. Another implementation technique also known as *LUT-based Serial Distributed Arithmetic (SDA)* contains a filter structure in which multipliers are no longer presented, resulting in an overall reduction of logic cells. This technique can be seen as *the minimum cell topology*. It's the topology which consumes the fewest amount of logic cells. Disadvantageously this topology needs a **Look-Up Table (LUT)** filled with 2^N coefficients whereas the traditional direct form I uses only N filter coefficients.

The LUT-based SDA Direct Form I with Noise Reduction structure consumes 551 logic cells and $64 \times 28 = 1792$ memory bits for coefficient storage. A timing analysis showed that the maximum clock frequency for this circuit is 32.89 MHz. Compared to the traditional topology with noise reduction, this solution reduces the amount of logic cells with 50%.

Whereas the traditional structure needs only an update of 5 coefficients per channel with a word length of 27 bits, the LUT-based structure requires an updating of 32 coefficients per channel of 28 bits. The LUT-based structure requires a lot more processing power of the controller (which calculates a new set of coefficients), and in addition a fast interface protocol from controller to the EPLD is necessary.

An overview of the implemented filter structures using an Altera FLEX 10K30⁸ is given in Fig. 13. It shows the

⁸The Altera FLEX 10K30 device consist of 1728 logic cells

amount of logic cells and memory bits which are consumed. Various registered performances were observed during the timing analysis of different programmable logic device families.

VII. MEASUREMENTS

The performances of the structures were measured with the *Audio Precision System Two*⁹. This audio measuring instrument can handle sample rates up to 48 kHz and word lengths of 24 bit. At this time the *Audio Precision System Two* is not able to handle signals with a sample rate of 96 kHz (as the updated version, the *System Two Cascade*, will do), therefore the implementation was restricted to 24 bit/48 kHz. This limitation has no significant consequence for the implemented structures [9].

The direct form I structure can have poor filter performance near $|z| = 1$ caused by the scarcity of the pole locations in this region. If the structures-under-test perform well in these regions, it will do even better in others. To compare the performance of the different filter topologies, a low-frequency, high-Q, cut-filter is implemented. The musical parameters of this filter are: $g = -60$ dB, $f_0 = 30$ Hz, $Q = 10$. A gain of -60 dB is chosen because the equalizer can end up in a cascade form, where the total gain can easily exceed 18 dB. The frequency resolution of the *Audio Precision* in the digital domain is restricted from 10 Hz to 22 kHz. By testing the structures at low frequencies and to create useful plots, the center frequency of the filter is set to 30 Hz. The selectivity of the filter is set to its maximum value of 10. The frequency response of this filter simulated with MATLAB is shown in Fig. 14.

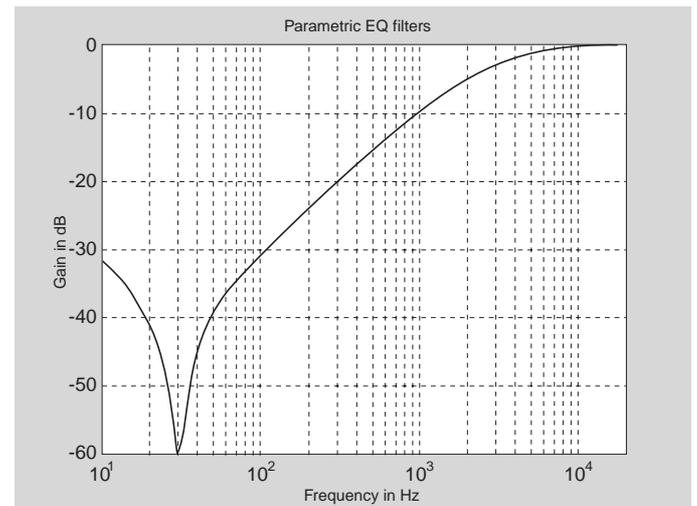


Fig. 14. Frequency response of a parametric equalizer simulated with MATLAB. The musical parameters are: $g = -60$ dB, $f_0 = 30$ Hz, $Q = 10$.

(also called logic elements (**LEs**)) and six EABs. These EABs results in a total of 12.288 RAM bits ($2,048 \times 1 \times 6$).

⁹Information about the *Audio Precision System Two* can be found at: www.audioprecision.com.

Altera Flex 10K30	memory bits	memory utilize (%)	logic cells	logic cells utilized (%)	FLEX 10K30 (MHz)	FLEX 10K30A (MHz)	FLEX 10K30E (MHz)
Traditional structures							
Direct form I	270	2	887	51	32.46	58.82	74.62
Direct form I with noise reduction	270	2	1,016	59	32.67	55.55	72.46
Direct form I with noise cancelation	270	2	1,348	78	32.05	54.05	72.46
Lut-based structures							
Direct form I	1,792	15	450	26	33.55	55.55	79.36
Direct form I with noise reduction	1,792	15	551	32	32.89	60.97	65.78

Fig. 13. Overview of the amount of logic cells and memory bits which are consumed by the different filter structures, as well as the maximum system clock which can be applied using various Altera FLEX 10K30 devices.

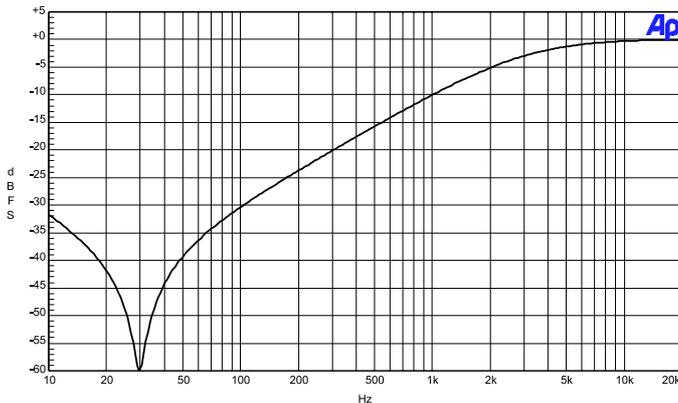


Fig. 15. Frequency response of a parametric equalizer implemented in an Altera FLEX 10K device. A 24-bit input signal at 0.00 dBFS and musical filter parameters, $g = -60$ dB, $f_0 = 30$ Hz, $Q = 10$, were used.

A. Direct form I

Fig. 15 shows the frequency response of the implemented filter.

A digital input signal of 24 bit/48 kHz at a level of 0.00 dBFS was generated by the *digital generator* of the *Audio Precision*. This signal was present at the input of the parametric equalizer while the frequency response at the output was measured with the *digital analyzer* of the *Audio Precision*. The frequency response of the filter was determined by sweeping a single sinewave, from 10 Hz to 20 kHz in 300 steps. The similarity of the frequency responses of the simulated (Fig. 14) and implemented filter (Fig. 15) curve is noticed.

Fig. 16 shows a parametric equalizer with the same musical parameters but this time the amplitude of the input signal was reduced to -50 dBFS. According to the theory described in [9], is observed that the amplification of the truncation error by the recursive part of the direct form I structure disturbs the frequency response of the filter.

B. Direct form I with noise reduction

In this subsection the direct form I structure was extended with a second-order noise reduction circuit. The frequency response of the parametric equalizer was tested using the same musical parameters: $g = -60$ dB, $f_0 = 30$

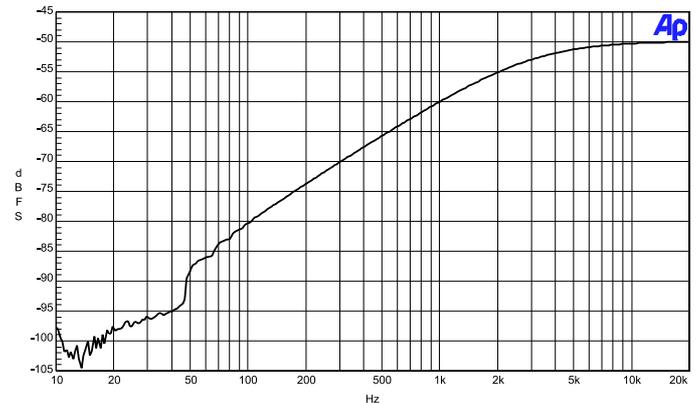


Fig. 16. Frequency response of a parametric equalizer using a input signal at a level of -50 dBFS, and musical filter parameters, $g = -60$ dB, $f_0 = 30$ Hz, $Q = 10$.

Hz, $Q = 10$. Even when the input signal was reduced to -50 dBFS the frequency response of the filter wasn't disturbed. Both filter responses of the direct form I and the direct form I with second-order noise reduction, are depicted in Fig. 17. Fig. 18 shows that even when the input signal was reduced to -80 dBFS the filter characteristic of the structure with a noise reduction circuit remained unchanged. From Fig. 18 it is noticed that the filter response of the traditional direct form I is even worse. It is also observed that the theoretical depth of the filter of -140 dBFS is not achieved. An explanation for this phenomenon is that regardless the filter under test boost or cuts it still add 6 dB truncation noise. Theoretically, using a 24-bit data path, we can approximately go as low as $6 - (24 \times 6) = -138$ dBFS.

VIII. SUMMARY

As it becomes obviously clear in [9], choosing a filter structure for a certain application is a complicated task. Although the direct form I filter has bad coefficient sensitivity effects and the truncation noise can be especially poor for poles grouped closely together and forced overflow oscillations can occur, it is the best candidate for an EPLD implementation. The coefficient sensitivity effects can be overcome using 24 bit filter coefficients and the truncation noise can be reduced using suboptimal ESS or can even

Direct form I with Noise Reduction	Traditional structure		LUT-based structure	
	4 mono channels	8 mono channels	4 mono channels	4 mono channels
Device Features				
logic cells	8,128	16,256	4,408	8,816
memory bits	2,160	4,320	14,336	28,672
EABs	4	4	8	16
PLD Family				
Altera	FLEX 10K250	APEX 20K600	FLEX 10K130	FLEX 10K250
	FLEX 10K180	APEX 20K400	FLEX 10K100	FLEX 10K180
Xilinx	VIRTEX XCV400	VIRTEX XCV800	VIRTEX XCV200	VIRTEX XCV400

Fig. 19. Overview of the implementation of a 4 mono channel or 8 mono channel mixing-module in different programmable logic device families. Each channel includes a lowpass and highpass shelving filter as well as two parametric equalizers.

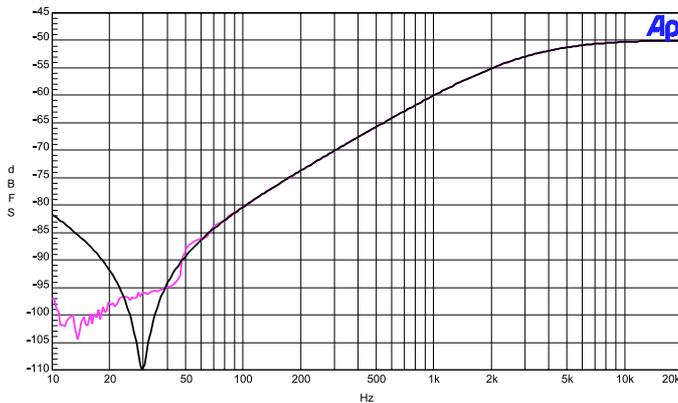


Fig. 17. Comparison of the filter response using a direct form I (grey line) and a direct form I structure with a second-order noise reduction circuit (black line). The amplitude of the input signal is -50 dBFS.

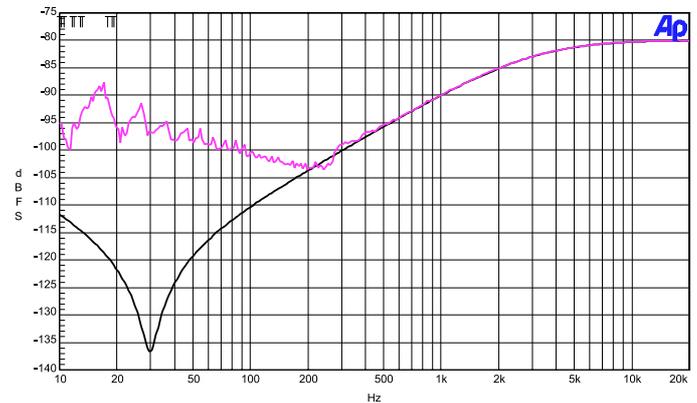


Fig. 18. Comparison of the filter response using a direct form I (grey line) and a direct form I structure with a second-order noise reduction circuit (black line). The amplitude of the input signal is -80 dBFS.

be cancelled using optimal ESS. At last, forced overflow oscillations can be eliminated using a saturator.

We can conclude that the Direct form I structure, or its LUT-based form with noise reduction appears to be the most elegant solution for an optimal EPLD implementation. This form realizes an elegant and optimal cell implementation whereas the filter performance is maintained.

If the parametric equalizer ends-up in a broadcast audio mixing-desk, a modular construction of the input channels would be advisable. For a modular construction, the mixing-desk can exist of several mixing-modules of 4 mono channels or 8 mono channels. Each mono input-channel of the mixing-module has to contain a lowpass shelving¹⁰ filter, two parametric equalizers for the mid range, and a highpass shelving filter.

Fig. 19 shows an overview of the consumption of logic cells and memory bits of the realization of a 4 mono channel or 8 mono channel mixing-module. Note that one filter structure already includes 2 mono channels. Fig. 19 also shows which PLD-families can be used for the imple-

¹⁰Shelving filters have adjustable gains and cutoff frequencies. The implementation of shelving filters can be accomplished using the filter structures mentioned in [9].

mentation of the mixing-module, applying traditional or LUT-based filter structures.

The application of an EPLD as a stand alone filter-core turns out to be an economical solution if we consider the implementation of a 4 or 8 channel mixing-module in various programmable logic devices, as summarized in Fig. 19, and it is also expected to be economical considering the costs. The solution presented in this report appears to be a real competitive alternative, building complex digital audio mixers.

It was shown that the present day EPLDs can fulfill complete DSP-functions, and that these devices are no longer restricted to perform simple data-format conversions. At this moment, the realization of a 8 channel mixing-module in an EPLD is a fact, and next year it will be possible to implement a **16 channel mixing-module in one EPLD**.

REFERENCES

- [1] R. Bristow-Johnson, "The Equivalence of Various Methods of Computing Biquad Coefficients for Audio Parametric Equalizers," presented at the 97th Convention of the Audio Engineering Society, San Fransisco, 1994 November, preprint 3609.
- [2] J. Dattorro, "The Implementation of Recursive Digital Fil-

- ters for High-Fidelity Audio," *J. Audio Eng. Soc.*, vol. 36, pp. 851-878 (1988 Nov.).
- [3] D. F. Elliott: *Handbook of Digital Signal Processing Engineering Applications*, Academic Press Inc., San Diego, California, 1987.
- [4] T. Laakso and I. Hartimo, "Noise Reduction in Recursive Digital Filters Using High-Order Error Feedback," *IEEE Trans. Signal Processing*, vol 40, pp 1096-1107, May 1992.
- [5] T. Laakso and J. Ranta, "Design and Implementation of Efficient IIR Notch Filters with Quantization Error Feedback," *IEEE Trans. on Instruments and Measurement*, vol 43, pp 449-456, June 1994.
- [6] J. D. Markel, A. H. Gray, "Fixed Point Implementation Algorithms for a Class of Orthogonal Polynomial Filter Structures," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-23 (1975 Oct.).
- [7] C. T. Mullis and R. A. Roberts, "An Interpretation of Error Spectrum Shaping in Digital Filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-30, pp. 1013-1015, Dec. 1982.
- [8] S.J. Orfanidis: *Introduction to Signal Processing*, Prentice-Hall, Englewood Cliffs, New Jersey, 1996.
- [9] M. Post: *Digital Audio Signal Processing Core: Realization of a Parametric Equalizer using an EPLD as a stand alone DSP-Filter-Engine*, University of Twente, Faculty of Electrical Engineering, Signals & Systems, Enschede, The Netherlands, Report Code: 029-99, August 1999.
- [10] R. A. Roberts, C. T. Mullis: *Digital Signal Processing*, Addison-Wesley, Reading, Massachusetts, 1987.
- [11] R. Wilson, "Filter Topologies," *J. Audio Eng. Soc.*, vol. 41, pp. 667-678 (1993 Sept.).
- [12] U. Zölzer, "Roundoff Error Analysis of Digital Filters," *J. Audio Eng. Soc.*, pp 232-244 (1994 April).
- [13] U. Zölzer: *Digital Audio Signal Processing*, John Wiley & Sons, Chichester, 1997.

This page was intentionally left blank