FPGA Implementation of Polyphase Decomposed FIR Filters for Interpolation Used in Δ-Σ Audio DAC

Noura Ben Ameur(1), Graduate Student Member, IEEE, Maher Soyah(2), Nouri Masmoudi(1), Senior Member, IEEE, Mourad Loulou(1), Senior Member, IEEE

(1)Electronics and Communication Research Group, Laboratory of Electronics and Information Technology Department of Electrical Engineering, National Engineering School of Sfax, Tunisia

(2) National Engineering School of Sousse, Tunisia

noura.benameur@ieee.org, soyah.maher@gmail.com, Nouri.Masmoudi @enis.rnu.tn, mourad.loulou@ieee.org

Abstract— **This paper describes a synthesis design from the MATLAB model into VHDL of a digital interpolation filter algorithm, used in a Δ-Σ Digital-to-Analog converter (DAC), intended for Professional digital audio system. The whole filter system simulation, VHDL implementation and Field Programmable Gate Array (FPGA) verification are processing. The Register Transfer Level (RTL) simulation result show an achieving a high resolution of a 22.5-bit at a switching rate of 8.192 MHz and a small area less than 50% of the total chip. Timing analysis indicates any violation of temporal constraints and the worst-case maximum clock speed of the design can attain a 500 MHz.**

Keywords—**Interpolation, polyphase decomposition, MATLAB, RTL simulation, Audio applications**

I. INTRODUCTION

The growing use of an audio digital-to-analog converter (DAC) in compact disks (CDs), DVDs, (Digital Video Disc), movie soundtracks, broadcasting and computer applications has increased [1] [2]. Recent designs mostly concern low power, but ignore the consideration of performance. As we know, there are several key specifications for a DAC intended for audio systems, except for the low power dissipation, high performance and small area are also required. For CD-quality audio signals, the signal to-noise-ratio (SNR) must also be at least 100 dB. Professional applications even require audio signals of >120 dB. This paper presents a 24-bit digital interpolation filter used in a Δ-Σ DAC. This design both consider the performance and area. The whole interpolation filter achieves a 22.5-bit in dynamic range (DR), which is sufficient for professional audio applications. At the same time, the polyphase [3] [4] [5] decomposed FIR filtering process optimized the digital filter's taps, that there are only 174 taps which greatly reduce the area. Those circuits operating at low sampling frequency fsi can particularly benefit from this optimization. A little changes of the taps can suit for the same type high resolution Δ-Σ DAC. The design is verified in Altera's Arria II GX FPGAs and tested in Quartus II software. The function and performance reach design target and have good prospect for the audio applications.

In this work, we consider the design of FIR filter to interpolate an audio input signal to a desired output sampling frequency. The alternative for use polyphase decomposed interpolation filters based on FIR filters, will be discussed when this alternative is advantageous in terms of exploiting the symmetry of the FIR filter to implement an optimal form of the filter bank, using resource sharing.

In the next section a discussion between two methods of digital filtering is shown. Then, in Section III we review polyphase decomposition of finite impulse response (FIR) filters and the resulting structure. In section IV FPGA implementation results of interpolation filter is presented. Finally, some conclusions are given in Section V.

II. INTERPOLATION FILTER DESIGN METHODOLOGY

Many digital systems use signal filtering [3] [6] to remove unwanted noise, to provide spectral shaping, or to perform signal detection or analysis. The question now arises if methods exist in which the interpolation process can be implemented practically in digital domain without adding significant distortion in the interpolated output [8]. Two types of filters that provide these functions are finite impulse response (FIR) filters and infinite impulse response (IIR) filters. Typical filter applications include signal preconditioning, band selection, and low-pass filtering. In contrast to IIR filters, FIR filters have increasing signal resolution, a linear phase and inherent stability. According to these characteristics a FIR filter will be used in the interpolation process for high resolution audio applications. However, for a given frequency response, FIR filters are a higher order than IIR filters, making FIR filters more computationally expensive. In this work, the performance of finite length interpolation filter method was chosen here and will be used to design and meet constraints according to audio signal specifications at the cost of greater computational complexity.

The optimum equiripple linear-phase FIR filter design method formulated as Parks-McClellan and Remez algorithms is viewed as an optimum design criterion [5][9]. The passband and stopband edges are at 28.8 kHz and 35.2 kHz, respectively.

978-1-4244-4398-7/09/$25.00 ©2009 IEEE

The filter is designed for a passband ripple of 0.00021 dB and a stopband ripple of -150 dB. A complete design of an interpolation filter is presented for high resolution digital audio.

Taking into consideration a hardware resource, that's to reduce the overall computational complexity of the designed digital FIR filter in section II, it can be implemented efficiently with polyphase filtering structure.

III. Polyphase Filtering

The interpolation process implements FIR polyphase filter structure to upsample a 24-bit audio signal sampled at the Nyquist Rate of 64 kHz to a sampling rate of 8.192 MHz. Resampling the signal in a single step (9628 taps) would require far more resources because the filter would be far more complex. This type of implementation is inefficient and should be avoided. Such a description would result in several million multiplications [3]. Interpolation of a signal by an integer oversampling factor can be accomplished by processing the input signal with the cascade of an oversampling factor and low-pass filter. However, larger stages of cascading and parallel processing may affect the designs execution time.

In a polyphase FIR filter structure, the interpolation low pass filter is divided into parallel sub-filters [3][4] and the filtering process is required in a low sampling rate f_{si}. Then, the frequencies are oversampled that are asynchronous to each other. Fig.1 presents the structural scheme of a polyphase interpolator, firstly OSR-1 sub-filters then zeros and finally delay-cells are inserted. The optimum equiripple linear phase filter method is the preferred filter design method. The search of the filter coefficients involved the use of the Matlab fdatool.

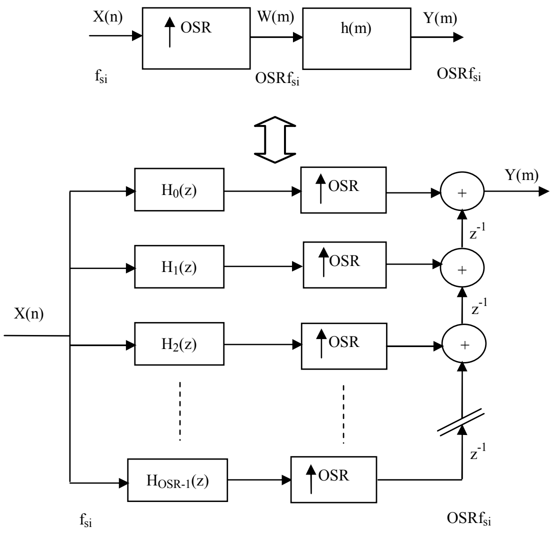

Figure 1. Polyphase structures for 1 to 128 interpolator.

We need to achieve 128 times interpolation, where no need for the (128-1) zeros insertion occurs, implying an efficient three-stage cascading polyphase filters [8] [10] with respectively oversampling factors (2, 4, 16) and optimizations based on symmetry [7] [8]. The total filter design tap count of 174 and an upsampling rate of OSR=128. The first one exhibit a length of 75 taps, the second, a length of 30 taps and the third a length of 69 taps. The whole structure of an audio interpolator is shown in Fig. 2. The Fig. 3 illustrates a zoomed view to see how the input signal has been interpolated.

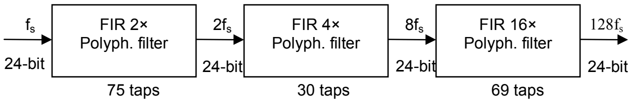

Figure 2. The whole structure of the Interpolator.

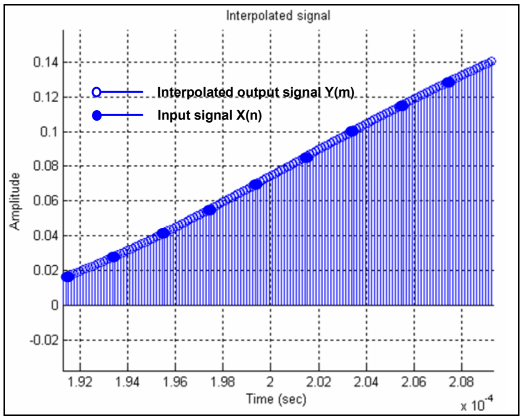

Figure 3. Input X(n) and interpolated output Y(m) of rate OSR=128.

In spectra level, Fig. 4 shows how the filter has preserved the cosine frequency components and has attenuated the frequency images below -150 dB which is 24-bit, the resolution of the input signal.

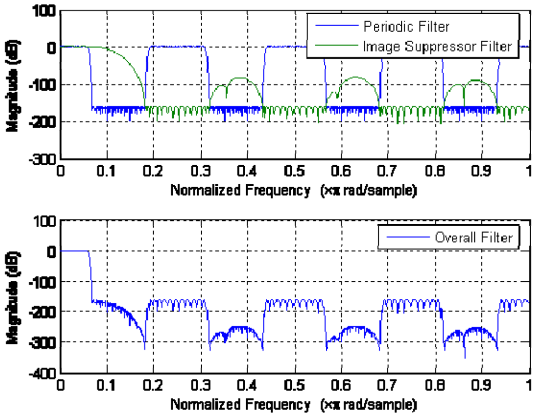

Figure 4. Magnitude response of the Interpolation chain.

The FPGA implementation of the interpolation process will be discussed in section IV, in order of the hardware resources occupation and temporal constraints verification.

IV. Implementation Results

In audio signal processing, a serial implementation of an audio algorithm is widely needed. The tools based on the filter design were MATLAB and Simulink from The MathWorks. Algorithms are defined using a special block set or description in the proprietary language and are later translated into an RTL hardware description language. The block set allows both single rate and multi-rate implementations. Fig. 5 shows the design flow using a FIR filtering process implementation proposed by ALTERA [10] [11] [12].

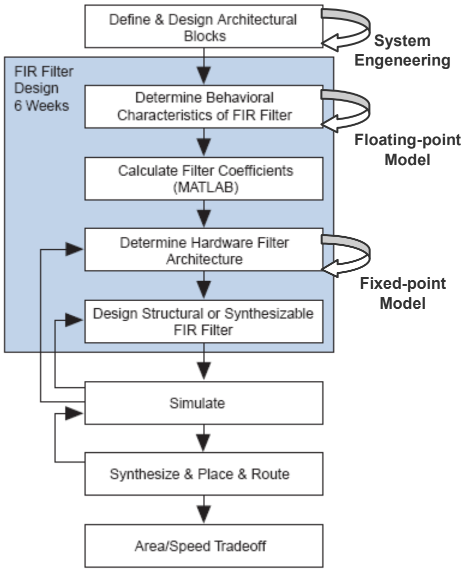
Define & Design Architectural Blocks
System Engeneering
FIR Filter Design 6 Weeks
Determine Behavioral Characteristics of FIR Filter
Floating-point Model
Calculate Filter Coefficients (MATLAB)
Determine Hardware Filter Architecture
Fixed-point Model
Design Structural or Synthesizable FIR Filter
Simulate
Synthesize & Place & Route
Area/Speed Tradeoff

Figure 5. FIR filter design flow.

An RTL description [11] [12] of the design was prepared in VHDL. Carry-skip arithmetic was used throughout. The Fig. 6 illustrates the implemented architecture. The current input variable Data_in is multiplied by all the coefficients of the sub-filters to produce the outputs. These multiple-output switch symbol to indicate the selection functions inside the respective circuits, then are added in the delay block to produce the filter output data_output.

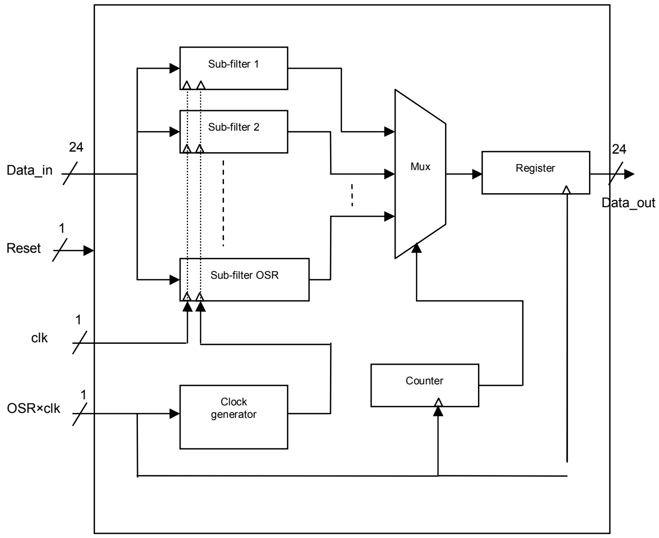

Figure 6. Implemented global architecture of Polyphase Interpolator.

The basic element which constitutes the interpolation filters are the sub-filters FIR with an order $N' = \frac{N+1}{0SR}$, N is the whole filter order. The Fig. 7 presents the developed architecture which implements on the one hand, the combinative arithmetic units carries out the basic arithmetic operations such as the addition and the multiplication, and on the other hand, the memory stacks (ROM and registers) ensuring the storage of the FIR filter coefficients.

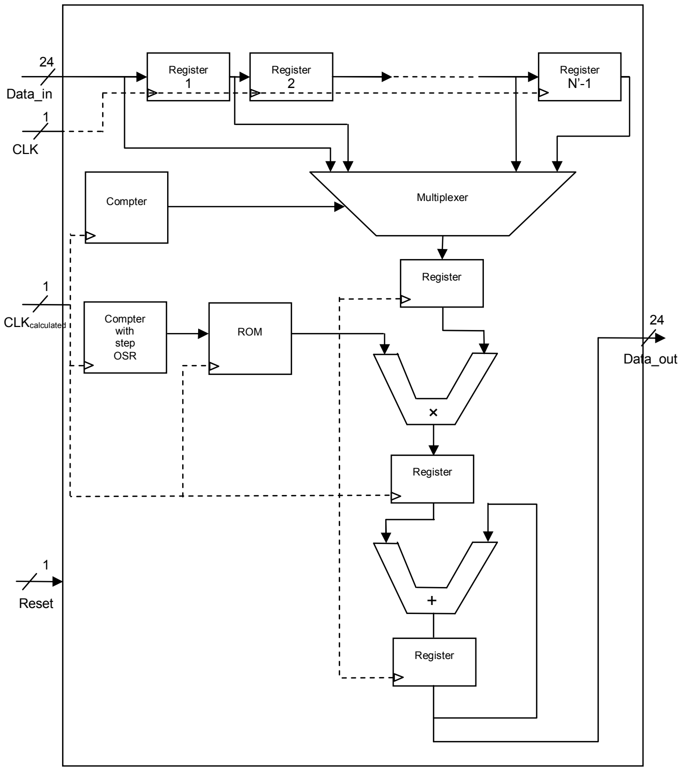

Figure 7. Implemented architecture of the sub-filter FIR.

A. VHDL achievement and Matlab simulation

The goal of the simulations here is to show how the ability of the interpolation audio data to maintain a 24-bit resolution within the audio baseband. Relative error result between MATLAB and VHDL simulations is shown in Fig.8. The Matlab results show the system's performance with quantized filter coefficients, but all internal computations with infinite precision. The VHDL simulation results include the finite precision of all computations. It is observed that the interpolation produces smoother interpolants and incurs a smaller error, for a 1.2-bit loss in DR due to the finite precision computations in the VHDL simulation across the entire bandwidth. The simulation results attained shows that high resolution interpolation can be practically implemented within an FPGA exhibiting a maximum clock speed of 500 MHz.

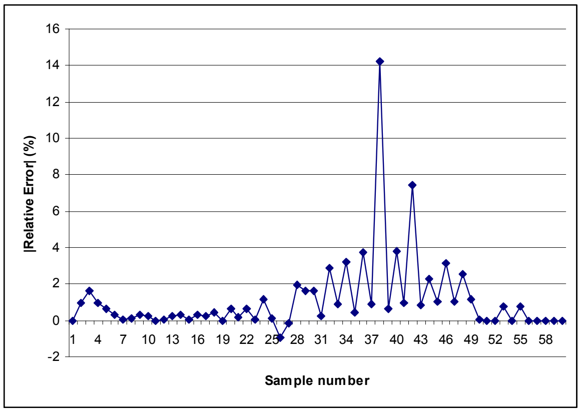

Figure 8. Relative error evaluation between MATLAB and RTL simulations of the interpolation filter.

B. Ressource utilization and Timing analysis

To evaluate how much of the Stratix FPGA resources are used, we synthesis the design in the Quartus II software, the FPGA chip we choose is Stratix GXII EP2SGX90FF1508C3 device of ALTERA constructor. Figure 9 shows the physical FPGA resource utilization for each interpolator. FPGA resource utilization is the measure of both the spatial and temporal allocation of the functional units, which is less than 50% of the total chip and is sufficient small in the digital system design. So, the filter function meets design requirements at a lower cost and with proven software support.

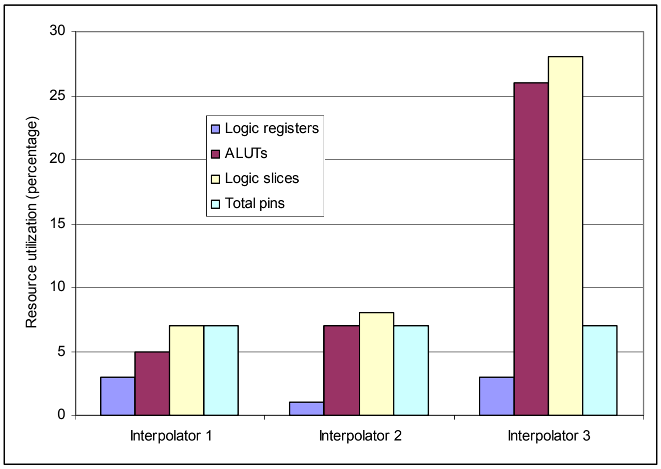

Figure 9. Resource utilization on the Stratix GXII EP2SGX90FF1508C3 FPGA.

Table I summarize the Timing analysis of the design. It can be observed that by comparing the maximum frequencies that can provide the FPGA and those required by the interpolation filters, one concludes that our model does not present any violation of the temporal constraints.

TABLE I. TIMING ANALYSIS OF INTERPOLATION FILTERS

		Interp. 1 (OSR=2)	**Interp. 2 (OSR=4)**	**Interp. 3 (OSR=16)**
FPGA device family		***Stratix GXII EP2SGX90FF1508C3***		
Maximum frequencies	clk×76/ clk×16/ clk×9	38.79 MHz	34.53 MHz	36.66 MHz
	clk×2/ clk×4/ clk×16	495.05 MHz	478.01 MHz	356.63 MHz
Required frequencies	clk×76/ clk×16/ clk×9	4.832 MHz	1.92 MHz	4.416 MHz
	clk×2/ clk×4/ clk×16	128 kHz	512 kHz	8.192 MHz

V. CONCLUSION

System-level considerations of polyphase decomposed interpolation filter used in Δ-Σ DAC and implementation have been presented. The design optimizes the taps, which saves the area and takes care of trade-off between performance and area, the degradation in DR is less than 1.2-bit.The whole design is verified in FPGA, the function of the filter is correct and the design is fully optimized, the area is small enough to ensure that the hardware performance metrics meet the system specifications for the audio applications.

REFERENCES

[1] I. Lokken, "High Resolution Audio DACs",Final Report, A Review of the digital Audio Conversion Process, Aout 2005.

[2] D. Reefman et. al. , "A 128fs, Multi-bit ΣΔ CMOS Audio DAC with Real-time DEM and 115 dB SFDR", Philips Semiconductors, Eindhoven, the Netherlands. AES Preprint 5846, March 2003.

[3] Alan V. Oppenheim and Ronald W. Schafer, "Discrete-Time Signal Processing", Prentice Hall, Upper Saddle River, New Jersey, second edition, 1999.

[4] D. George, "A Firmware-Based Polyphase Filter Design Tool", Mémoire de fin d'études, Univeristé de Cape Town(Afrique du Sud), Département de Génie-Electrique, Octobre 2005.

[5] P. Gomez, "A Practical Method to Design Equiripple FIR Filters", Florida International University, Department of Electrical and Computer Engineering, Novembre 2001.

[6] M. Cerna, A. F. Harvey, "The Fundamentals of FFT-Based Signal Analysis and Measurement", Application Note, National Instruments, Juillet 2000.

[7] Phil Schniter, "Computational Savings of Polyphase Interpolation/Decimation", Version 2.2, Sep 20, 2005.

[8] N. Ben Ameur, M Loulou, "Design of Efficient Digital Interpolation Filters and Sigma-Delta Modulator for Audio DAC", IEEE International Conference on Design &Test of Integrated Systems (DTIS'08), March 25-28, 2008 Tozeur, Tunisia.

[9] N. Ben Ameur, M. Loulou, "A 24-Bit, 8.1-MS/s D/A Converter for Audio Baseband Channel Applications", International Journal of Electronics, Circuits and Systems (IJECS), 2008.

[10] J. Stephenson, "Design Guidelines for Optimal Results in FPGAs", Altera Corporation, 2005.

[11] V. A. Pedroni, "Circuit Design with VHDL", MIT press, 2004.

[12] P. P. Chu, "RTL Hardware Design Using VHDL: Coding for Efficiency, Portability, and Scalability", Wiley-Interscience publication, 2006.