Multirate IIR Filter approach for reduce energy and silicon consumption

Omitted due to the blind review process
Dept. name of organization
Name of organization
City, Country
E-mail: name@xyz.com

Omitted due to the blind review process
Dept. name of organization
Name of organization
City, Country
E-mail: name@xyz.com

Abstract— **The growing increase in density of integrated circuits propitiates the emergence of systems embedded in a single chip more complexes. This growing on the electronic circuits significantly influences on the growing of the energy consumption, making the search for minimizing of this consumption a constant concern on conception of new devices. To reach this goal, is used the technique of design review, focusing on the reduction of silicon consumption and on the reduction of operation frequency of the circuits. The presented work has done a evaluation of a telecommunication application that uses this approach, exploring the design of a multirate filtering process, implemented originally using the FIR filter topology, and suggesting the use of filter with IIR topology. The results meet the requirements of energy consumption, as it will be shown.**

FIR, IIR, FPGA, Multirate.

I. INTRODUCTION

The growing increase in density of integrated circuits and the consequent reduction on silicon cost has enabled designers to implement complex systems in a single silicon chip, commonly known as Systems-on-Chip (SoCs) [1]. So, new applications emerge with more complexity in a single chip. To meet this demands, in recent decades were available some integrated circuits that may have its electronics circuits programmed by the designers. They are programmable logic devices with high density, known as FPGA (Field Programmable Gate Array). The uses of this kind of devices, allies to efficient tools used to synthesize the circuits, has popularized the uses of the FPGA in the industry.

Complex integrated circuits naturally demands by a greater energy consumption. Therefore, parallel to the growing increase in density of integrated circuits, more and more the designers should be concerned to implement designs that are efficient and that demand for less energy. Reviews the methods used to solve some technical problem in a design or reduce the processing load in the system are the first's alternatives chosen.

The work presented on this paper is inserted on this context. It shows an alternative technique that offer a reduction on the silicon consumption and minimizes the processing load for an application used in telecommunications.

The application studied is a first order delta-sigma codec that uses a multirate system to suit the sampling frequency of the modulator, fixed as 2048kHz, to the operation frequency adopted in telephony, defined as 8 kHz. The first work, that originated this codec, was presented by [2]. In that work was adopted the uses of FIR filters (Finite Impulse Response) to perform the decimation and interpolation on this codec, that later was implemented in a FPGA by the company Intelbras S.A. The purpose of this work is the replacement of that FIR filters by IIR filters (Infinite Impulse Response). The IIR filters, when designed for a same Passband and Stopband specifications that the FIR filters, needs less silicon consumption and a lower quantity of arithmetic operations to do a filtering process.

The codec was implemented in a FPGA and its silicon consumption and number of operation cycles were accounted and compared with each other, for the implementation of decimators and interpolators using the FIR and IIR techniques. The Sections that follows shown some studies related to this work, describes the implementation techniques that were used and the results obtained.

II. RELATED WORKS

Is usual the description of the multirate filtering process using FIR filters. Generally it is done due to the simplicity to implement a FIR filter rather than IIR filters and, also, because the FIR filters are stable, which may not occur with the IIR filters, depending to the behavioral of the application on that the filters are inserted.

Telecommunications applications that work with voice have consecutives samples of a signal strongly correlated and the stability problems that are related to the stability on use of IIR filter are minimized. On this Section are presented three works that are related to the approach of use of IIR filters to implement a multirate filtering.

The first related work is a telecommunication application presented by [3]. On that work, is described the design of a delta-sigma codec that is used on mobile telephony, on which the A/D converter (Analog-to-Digital) uses a linear word with 13 bits and sample frequency of 8 kHz. The Passband of that application was between 300Hz and 3.4 kHz. The inferior Stopband was specified to have -20 dB for frequencies below 100 Hz, -10 dB on the region between 100 and 200 Hz. The superior Stopband was specified to have -30 dB no the region between 4.0 kHz and 4.6 kHz and -40 dB for frequencies above 4.6 kHz. On that work, the multirate treatment needed to the decimation and to the interpolation are performed with the support of two specific purpose processors, designed to that application.

The second related work was presented by [4] and uses as application a device made by manufacturer STMicroeletronics®, called STLC7545. It is an integrated circuit used as analogue front-end to modems that meet the CCITT standard and works at rate of 38 to 400 bits per second. This device also uses a delta-sigma codec to perform *the A/D and D/A (Digital-to-Analogue) conversion process.* The decimation and interpolation processes were implemented with three IIR filters, with 14th order each. According those authors, the filter design was performed with the support of the tool MATLAB®.

The third related work is the implementation of a first order delta-sigma modulator to be used on the transmition and reception of analogue signals on equipments like PBX e was presented by [2]. On that work, the sampling rate of the analogue signal is 2048 kHz and the digitalized signal is sent to a TDM bus, with 30 channels. On the specification of that codec, were used as reference the standard to voice treatment employed by commercial codecs, like MT8960 and Le58QL020, both made by the manufactures Zarlink®, and that are used for analogue-Digital-Analogue conversion in PBX equipments. For that codec, the Passband was established between 300 Hz and 3.4 kHz and the anti-aliasing filters were specified as a low pass filters, with Stopband between 4.0 kHz and 4.6 kHz and with gain of -32 dB for frequencies above 4.6 kHz. On those devices, according the Zarlink ® is expected that the psofometric noise not exceeding the limit of -83 dB. The digitalized samples on that codecs are available on a TDM bus, as 8 bit words, compressed according the G711 CCITT standard.

On the Section that follows, the implementation of the IIR filters suggested are presented as alternative with low silicon consumption and low energy consumption for the implementation presented by [2], due to the lower number of coefficients and to the lower arithmetic operations needed for this filter topology. The details of the modulator, codec G711 and other hardware structures that were implemented in VHDL by the company Intelbras S.A. will not be showed on this paper, keeping focused on the implementation of the decimators and interpolators filters.

III. IMPLEMENTATION OF THE SYSTEM

A. Macro architecture of the system

The definitions used on this work are the same that were used by [2], like the sampling rate that has to be compatibilized are 2048 kHz and 8 kHz. Therefore, the multirate system that has to be developed should perform the decimation with a factor of 256, between the A/D converter and the TDM bus and an interpolation with the same factor.

The same way as was done on the work presented by [2], this work used three stages for decimation and three stages for interpolation. The Figures 3 and 4 shows the different stages that were used. Were applied decimation factors of 16, 8 and 2 for the stages 1, 2 and 3, respectively. And, for the interpolation, were applied the factors 2, 8 and 16 for the stages 1, 2 and 3, respectively. The utilization of the anti-aliasing filters for the decimation and for the interpolation also is shown in the Figures 1 and 2.

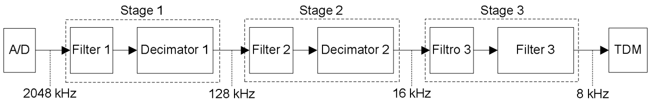

Figure 1. Block diagram representing the multiples stages of decimation with anti-aliasing filters.

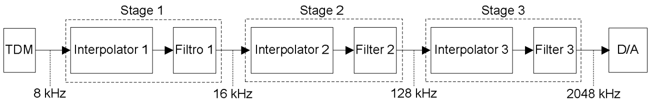

Figure 2. Block diagram representing the multiples stages of interpolation with anti-aliasing filters.

All the system was developed using VHDL language to be embedded in a FPGA XC3S500PQ208, from the manufacturer Xilinx®. This system works with signals between 300 Hz and 3.4 kHz and is digitalized with words 16 bits size. The chosen of the word with 16 bits was done to ensure the psofometric noise that is guaranteed by the commercial devices MT8960 and Le58QL020 also be guaranteed by the delta-sigma codec.

B. Filter design

The software tool called FDATOOL® (Filter Design & Analysis Tool), distributed with MATLAB®, was used on this work as a support tool to find the value of the coefficients and the quantity of coefficients needs for each IIR filter,

The cutoff frequency was defined with 200 Hz above the greater voice frequency. The reason is offer a smooth slope, needing a lower quantity of coefficients. The sampling rate and the stop frequency were done according [5]. The attenuation value of the stop frequency was defined with -32 dB due this value be adopted for the commercial codecs that were mentioned.

Using these specifications, the FDATOOL® calculated the order of each filter. The orders calculated were 2, 3 and 8 for the stages 1, 2 and 3, respectively. The orders calculated for the interpolators filters were 8, 4 and 2, for the stages 1, 2 and 3, respectively. The Tables 1 and 2 shows a resume of these specifications.

TABLE I. SPECIFICATIONS FOR THE DECIMATORS FILTERS.

	Filter 1	Filter 2	Filter 3
Sampling frequency (kHz)	2048	128	16
Cutoff frequency (kHz)	3.6	3.6	3.6
Stop frequency (kHz)	124	16	4
Attenuation on the Stopband (dB)	32	32	32
Word size (bits)	16	16	16
Gain on Passband (dB)	0	0	0

TABLE II. SPECIFICATION FOR THE INTERPOLATORS FILTERS.

	Filter 1	Filter 2	Filter 3
Sampling frequency (kHz)	16	128	2048
Cutoff frequency (kHz)	3.6	3.6	3.6
Stop frequency (kHz)	4	8	8
Attenuation on the Stopband (dB)	32	32	32
Word size (bits)	16	16	16
Gain on Passband (dB) (dB)	3	18	40

The Figure 3 shown the diagrams obtained from FDATOOL® for the decimators filters presented on Table 1. Similar diagrams were done for interpolator filters and are shown in Figure 4.

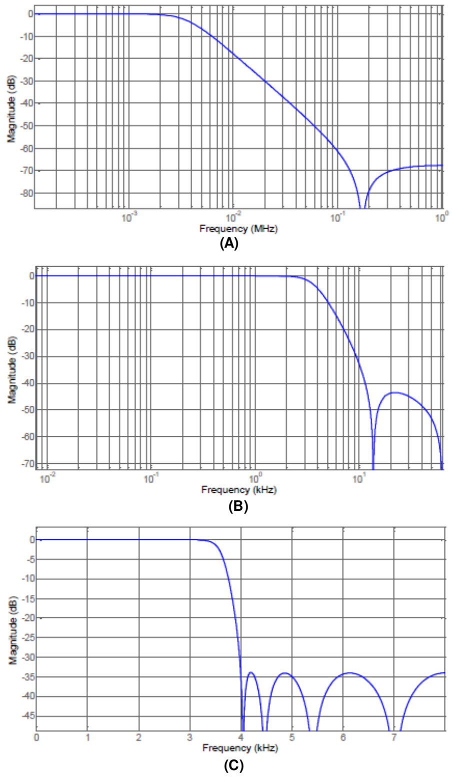

Figure 3. Diagrams for decimator filters: (A) first stage, (B) second stage and (C) third stage.

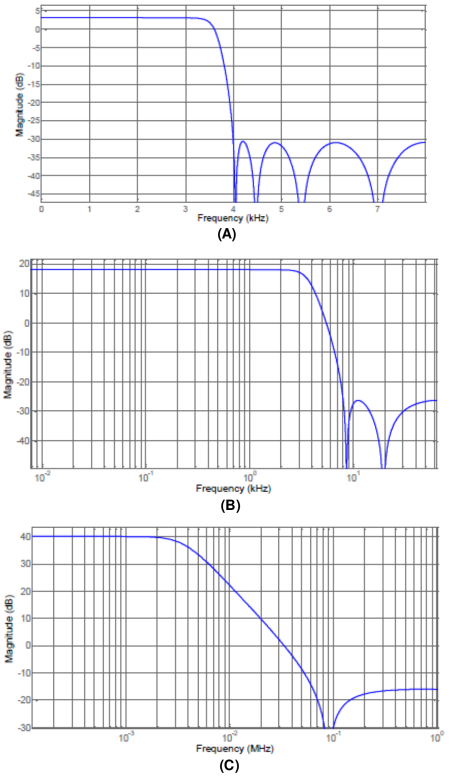

Figure 4. Diagrams for interpolator filters: (A) first stage, (B) second stage and (C) third stage.

C. *Decimation process*

To implement each decimation stage, shown in Figure1, were designed using VHDL language a structure constituted by two blocks of RAM memory, a filtering structure and a state machine. This state machine controls the flow of data through the decimation block and all decimation stage use a similar block. The Figure 5 shows a block diagram with these described structure.

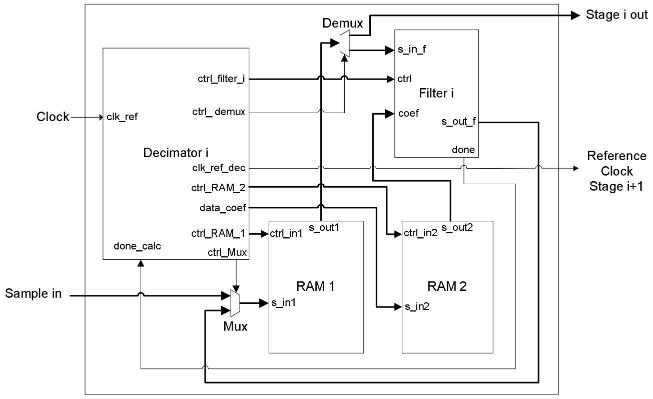

Figure 5. Block diagram showing the internal structure of a decimation stage.

According to the Figure 5, the diagram consists by six components: a decimator I, a filter I, the RAM 1 and RAM 2, a MUX and a DEMUX. The decimator block controls all other components. Its behavior was implemented as a finite state machine, shown in Figure 6. The block RAM 1 stores the input samples and the past samples, needed to IIR filtering process. The block RAM 2 stores the coefficients needed to the filtering process. The selections of which data will be stored or read are done by the MUX and DEMUX, duly controlled by the state machine.

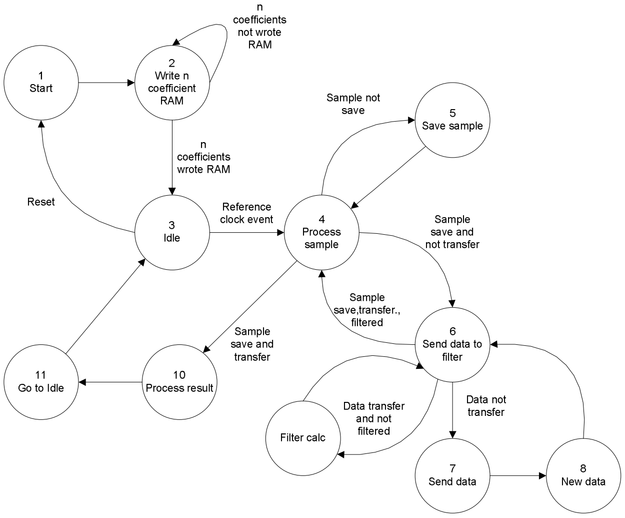

Figure 6. Finite state machine used to control the decimation or interpolation.

D. Interpolation process

The interpolation stage was implemented similarly to the implementation that was done to the decimation stage. To each interpolation stage, showed in Figure 2, was created a bock diagram, with a structure very close to the decimator block. The main differences are in the state machine that has to manipulate the samples differently when it works on interpolation or decimation process. The interpolation stage was implemented similarly to the implementation that was done to the decimation stage. To each interpolation stage, showed in Figure 2, was created a block diagram, with a structure very close to the decimator block. The main differences are in the state machine that has to manipulate the samples differently when it works on interpolation or decimation process and the MUX used with three inputs. The Figure 8 shows the interpolation block.

One of the differences in the interpolator block is that there is no a signal to indicate that the interpolation was finished. Another difference is that the MUX is able to connect to its output a value zero, used by the filter to do the filtering process interpolating zero value samples.

The finite state machine showed in Figure 6 is used to control the interpolation stages. They differ only in two aspects. The first aspect is that the state machine inserts zeros into the interpolation filter. In decimation stage, the state machine inserts samples from the input into the decimation filter. Another difference is that the state machine allows the interpolated value to be available on the output at every clock cycle, while in the decimator the output is available only every M clock cycles.

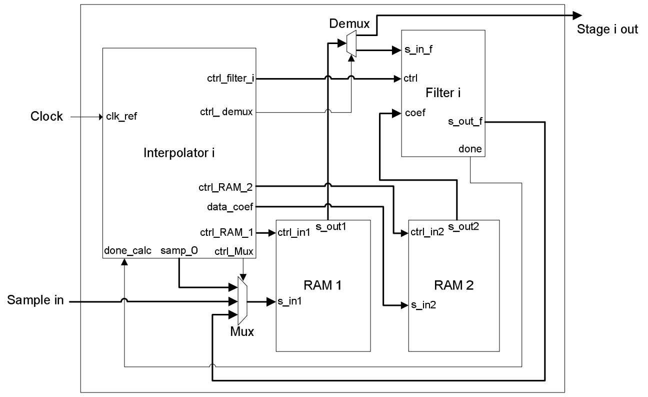

Figure 7. Block diagram showing the internal structure of a interpolation stage.

IV. Metrics and obtained results

This work adopted two metrics, proposed by [5], used to evaluate multirate systems: (*i*) the number of multiplications used to calculate an output value; (*ii*) the quantity of memory used to store the data needed in the filtering process and.

The first metric is the number of multiplications executed per second (MPS), and it is calculated as shown in (1):

$$MPS = \sum_{\mathrm{i}=1}^{\mathrm{k}} \mathrm{CiFi} \qquad (1)$$

Where *i* is the stage of the multirate system, *k* is the number of stages of the mutirate system, *Ci* is the number of coefficients of the *i* and *Fi* is the sampling frequency on the stage.

The expression presented in (1) is not valid to calculate MPS to multirate systems that use IIR filter in the decimation process, because this expression consider the sampling frequency at the of the output stage, but the decimation process happens before the reduction of the sampling rate. For FIR filters it is valid, because the result depends only on present samples, the filter needs perform the filtering process only at N clock cycles, resulting on the sampling frequency of the decimation stage. On the other way, in the IIR filters, the filtering process depends of the past samples. Therefore, to perform the MPS for an IIR multirate filter frequency in a decimation process should be used (2):

$$MPS = \sum_{\mathrm{i}=1}^{\mathrm{k}} \mathrm{NiF}i\_n \qquad (2)$$

Where *Fi* is the frequency at the input of the stage.

For the interpolation process, both filters topology may use (1) to calculate the MPS.

The other metric used on this work, refers to memory storage capacity, called TSR (Total Storage Requirements), needed to store the data from filtering process. According [5], the TSR may be calculated as shown in (3):

$$TSR = \sum_{\mathrm{i}=1}^{\mathrm{k}} \mathrm{Ci} \qquad (3)$$

This expression defines that the memory capacity needed to perform a filtering process in a multirate system is the sum of the number of coefficients of all filter used in the system, but this equation do not considers the possibility to store past samples for the IIR filters. On this case, should be adopted the (4):

$$TSR = \sum_{i=1}^{k} 2Ci \tag{4}$$

The quantity of coefficients to an IIR filter is obtained by (5):

$$C = 2N + 1 \tag{5}$$

Where *N* is the filter order. The Table 3 shown a comparison of MPS between the multirate system implemented com FIR filter and the multirate system implemented with IIR filter considering the decimation process, interpolation process and the sum of both.

TABLE III. COMPARISION OF MPS FOR FIR AND IIR MULTIRATE SYSTEMS.

	FIR - MPS ($x10^3$)	IIR - MPS ($x10^3$)
Decimation	8336	11408
Interpolation	227504	11664
Total	235840	23072

The Table 4 shown the TSR metric between the multirate systems implemented with FIR and IIR filters, considering the decimation, the interpolation and the sum of both.

TABLE IV. COMPARISION OF TSR FOR FIR AND IIR MULTIRATE SYSTEMS.

	FIR - MPS	IIR - MPS
Decimation	400	58
Interpolation	550	62
Total	950	120

The design initially described in this paper implemented just a bidirectional channel to transmit and receive voice in a TDM bus. Though, is possible to extend the results for cases with more channels, just replicating the previous results. Lets simulate a case where the systems were designed to work with more channels, ten for instance. The Table 5 illustrates this implementation with more channels in the FPGA chosen, considering that a bi-directional channel needs the interpolation and decimations process, using the values showed in Table 3 and 4 and multiplied by the quantity of channels.

TABLE V. COMPARISION OF TSR AND MPS FOR A IMPLEMENTATION WITH TEN CHANNELS

	MPS ($x10^3$)	TSR
FIR	2358400	9500
IIR	230720	1200

V. CONCLUSIONS

The numbers presented in this paper shown that a multirate system with multi stages when implemented using IIR filters, considering the same parameter of a same design implemented with FIR filters, presents computational gain in relation to the multirate systems implemented with FIR filters, as expected. For a bidirectional system, on which occur a complete decimation process and other complete interpolation process, the topology with IIR filter uses just 9.8% of the MPS that was necessary to a system using FIR topology, minimizing, therefore, the energy consumption of the system. Another advantage that was confirmed is that the system implemented with IIR filters uses 12.6% of the memory that was necessary to the system using FIR filter. Another important consideration is that the IIR filter may be designed to be more aggressive in relation to stop band levels, for instance, with a lower resources consumption that is necessary for a FIR filter. This may be important to adjust the codec when it is been used in a specific field situation.

ACKNOWLEDGMENT

Our thanks to Intelbras Company for his kindness in giving your project coded delta-sigma as a case study.

REFERENCES

[1] Benini, L.; De Micheli, G., “Networks on chips: a new SoC paradigm,” IEEE Computer, v. 35, n. 1, p. 70-78, Jan. 2002.

[2] Berejuck. M., “Study and Implementation of Sigma-Delta codec for treatment of voice in telecommunications using FPGA,” 2003. 75p. Federal University of Santa Catarina. PPGEEL, Florianópolis, Brazil.

[3] Morling, R.C.S.; Kale, I.; Tsang, C.W.; Morris, S.J.; Hague, G.; Foran, C.; , "The design of a sigma-delta codec for mobile telephone applications ," Advanced A-D and D-A Conversion Techniques and their Applications, 1994. Second International Conference on , vol., no., pp.11-17, 6-8 Jul 1994.

[4] Ikram, M.Z.; Siddiqui, A., "Multirate IIR filter design and implementation in STLC7545 analogue front-end for voice communication," Electronics Letters, vol.34, no.1, pp.32-33, 8 Jan 1998.

[5] Ifeachor, E. C.; Jervis, B. W., “Digital signal processing: a practical approach,” 2 ed. Edinburgh Cabe: Prentice Haal, 2002.