Advances in Electrical and Computer Engineering **Volume 10, Number 2, 2010**

Conceptual Implementation of Sample Rate Convertors for DACs

Gabriel ANTONESEI[1], Cristina TURCU[2], Adrian GRAUR[3]
[1]Analog Devices Inc. USA, [2]"Stefan cel Mare" University
[1]804, Woburn Street, USA-01801 Wilmington, MA,
[2,3]str.Universitatii nr.13, RO-720229 Suceava
[1]gabriel.antonesei@analog.com, [2]cristina@usv.ro, [3]adrian.graur@usv.ro

Abstract **— One of most common and difficult challenge when creating a single SoC with digital (sub)sections is caused by the various master clock (MCLK) frequencies that each individual IC had originally. There are several methods to solve this, but when constraint by price and power consumption, the design engineers must find the optimum one. The sample rate converters (SRC) are an example of solution that can simplify the architecture in some of these cases. However, even for the SRCs themselves, we need to come up with novel and efficient architectures. This paper presents such an example from mobile phones chips on how to successfully mix on the same silicon, an audio sigma-delta DAC which should support all the standard audio rates using a 13MHz MCLK frequency imposed by the RF section incorporated inside the same chip. The document will go from showing the top-level digital signal processing down to the actual hardware implementation.**

Index Terms**— DAC, CIC, mobile phone audio subsystem, multirate filtering, sample rate converter**

I. INTRODUCTION

This paper deals with some elements related to the conceptual implementation of the fractional decimation and interpolation. This operation is also known as Sample Rate Conversion (SRC) because the input signal coming at a certain sampling frequency is changed into the same signal but with different sampling frequency. SRC is often applied in the mobile phone industry. In this paper a conceptual implementation of SRC for DAC channel of a mobile phone is presented (Figure 1, yellow block). Basically the SRC block has the next general description of the performed function: if a signal is received at the input with the sampling frequency Fs_in, it should be converted to the same signal, but sampled at the frequency Fs_out, while a certain amount of information, usually in a limited band, must be preserved.

Since the signals that we are dealing with are in the digital domain, it makes sense that the techniques to perform the SRC are implemented in the digital domain, as well.

Yet, this process can be understood easier using the idea of re-sampling after reconstruction: an analog signal is (virtually) reconstructed from the digital signal by means of D-to-A conversion and filtering. Eventually the reconstructed signal is re-sampled with the desired frequency and a new digital signal is obtained by means of A-to-D conversion. From this idea, the all-digital description of the process can be obtained keeping in mind that SRC is a process of re-sampling. The fundamental effects of sampling, imaging and aliasing, must be expected to appear with SRC [1].

This paper is based on some basic consideration for signal processing solutions used in sigma-delta based ADC and DAC converters that were presented in [2].

Fs_out is the working frequency of the DAC and Fs_op is the operating frequency of the SRC module. The sampling rate conversion ratios are presented in Table 1.

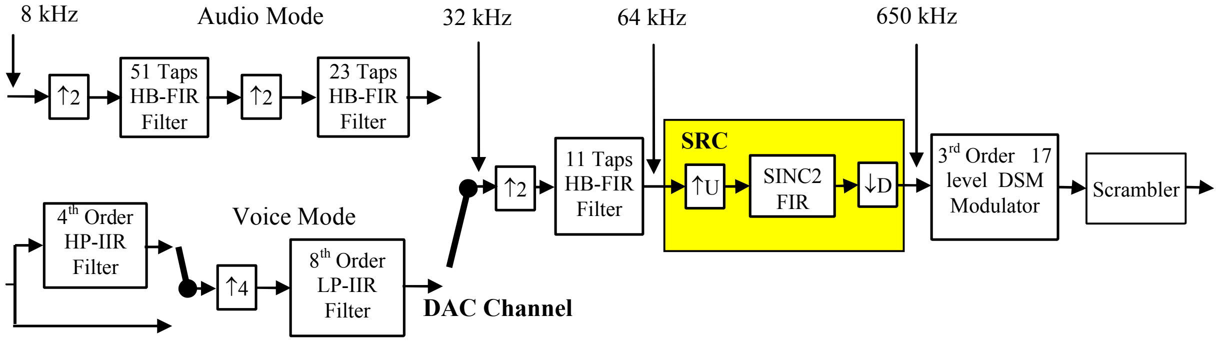


Figure 1. Digital section of the audio sub-system in a mobile phone – DAC path.

Table 1. DAC channel and the sampling rate conversion ratios (U-up rate, D-down rate).

Fs_in (kHz)	Fs_out (MHz)	U	D	Fs_op=U·Fs_in= =D·Fs_out
64	0.65	650	64	41.6
128	1.3	650	64	83.2
256	2.6	650	64	166.4
88.2	0.65	3250	441	286.65
176.4	1.3	3250	441	573.3
352.8	2.6	3250	441	1146.6
96	0.65	650	96	62.4
192	1.3	650	96	124.8
384	2.6	650	96	249.6

Digital Object Identifier 10.4316/AECE.2010.02009

Figure 2 shows the solution chosen to implement the SRC for the DAC channel:

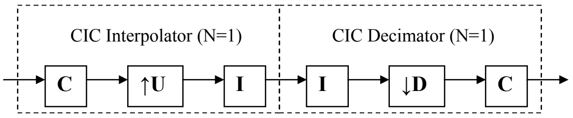


Figure 2. First order CIC interpolator concatenated with first order CIC decimator.

Using the techniques presented in [1], an equivalent diagram of the SRC module that uses moving-average (box-car) filters working at Fs_op frequency is obtained (Figure **3**):

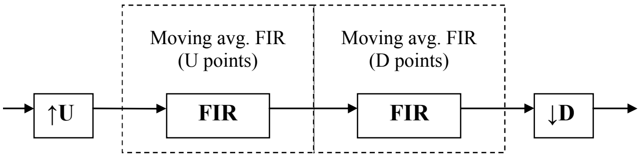


Figure 3. Equivalent diagram of concatenated CIC filters.

The transfer function of the SRC block in the z-domain and the normalized frequency-domain, referenced to the operating frequency (Fs_op) is:

$$H(z)=\left[\sum_{k=0}^{U-1} z^{-k}\right]\cdot\left[\sum_{k=0}^{D-1} z^{-k}\right] \tag{1}$$

$$\left|H(f)\right|=\frac{\left[\sin(\pi fU)\right]\left[\sin(\pi fD)\right]}{\left[\sin(\pi f)\right]^{2}} \tag{2}$$

The next several plots (Figure 4 - Figure 9) will present the zoomed frequency responses for the values U and D presented in the Table 1.

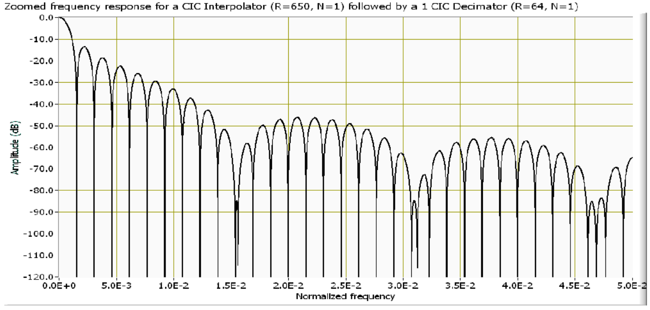


Figure 4. Zoomed frequency response for U=650 and D=64**.**

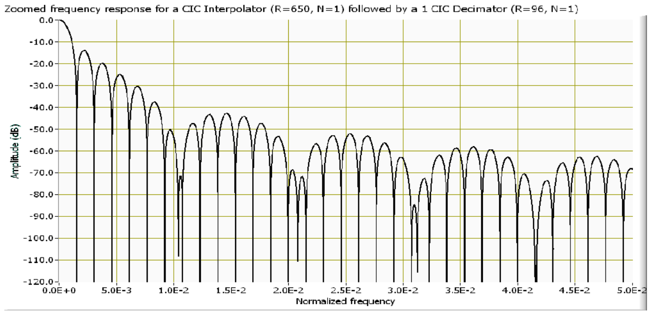


Figure 5. Zoomed frequency response for U=650 and D=96.

The following formula can be used to find the amplitude expressed in dB for the imaginary parts associated with a given tone:

$$\mathrm{Im}(k)=20\log\left|\frac{\sin(wU)\sin(wD)}{UD(\sin(w))^{2}}\right| \qquad w=\pi\frac{mk\pm 1}{mU},\quad k=1,2,\ldots,[U/2] \tag{3}$$

where:
U = upsampling rate; D = downsampling rate
m = normalized frequency of the tone referenced to Fs_in

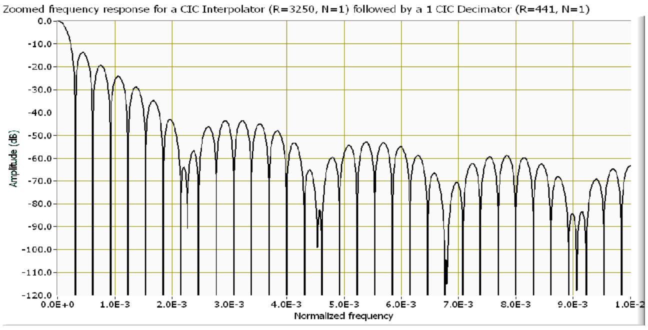


Figure 6. Zoomed frequency response for U=3250 and D=441.

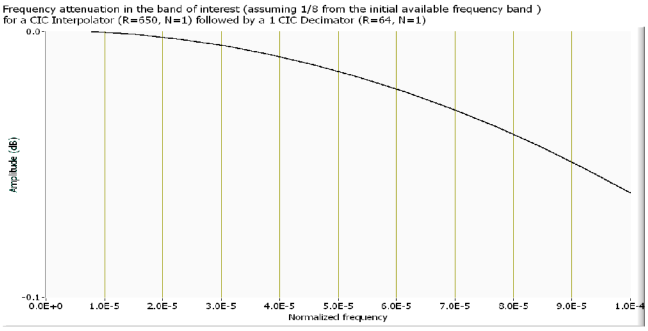


Figure 7. Frequency response in band of interest for U=650 and d=64.

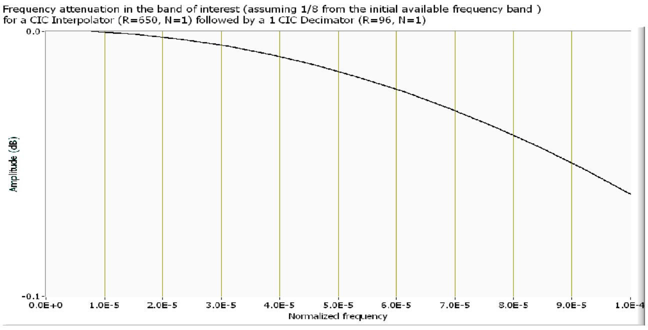


Figure 8. Frequency response in band of interest for U=650 and D=96.

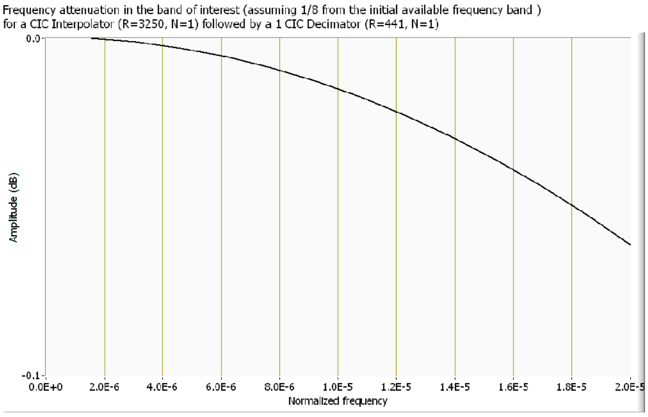


Figure 9. Frequency response in band of interest for U=3250 and D=441.

In Table 2 there are the first 5 imaginary parts (worst cases) when m is 0.05 (0.4 initially and then an increase by 8 of the sampling frequency before the SRC).

Observation: an algorithm that uses only a two stages CIC interpolator and no CIC decimator, will improve the attenuation of the imaginary parts (for example, the first ones will go below –48dB). The passband attenuation will increase a bit, instead.

Table 2. DAC-SRC: attenuation of imaginary parts in worst case scenarios.

	U=650, D=64		U=650, D=96		U=3250, D=441	
k	Im_left (dB)	Im_right (dB)	Im_left (dB)	Im_right (dB)	Im_left (dB)	Im_right (dB)
1	-25.74	-26.63	-25.89	-26.83	-25.85	-26.77
2	-32.39	-32.88	-33.08	-33.64	-32.88	-33.43
3	-36.69	-37.07	-38.36	-38.86	-37.88	-38.34
4	-40.27	-40.61	-43.55	-44.10	-42.57	-43.05
5	-43.65	-43.99	-49.69	-50.41	-47.76	-48.34

The following formula specifies at what positions will alias the imaginary parts (everything is referenced to Fs_op):

$$Fa = \begin{cases} remainder\left(\frac{k \pm Ft}{U}, \frac{1}{D}\right), \\ \quad if \;\; remainder\left(\frac{k \pm Ft}{U}, \frac{1}{D}\right) < \frac{1}{2 \cdot D} \\ \frac{1}{D} - remainder\left(\frac{k \pm Ft}{U}, \frac{1}{D}\right), \\ \quad if \;\; remainder\left(\frac{k \pm Ft}{U}, \frac{1}{D}\right) > \frac{1}{2 \cdot D} \end{cases} \quad (4)$$

Ft = normalized frequency of the initial tone; k = imaginary part index.

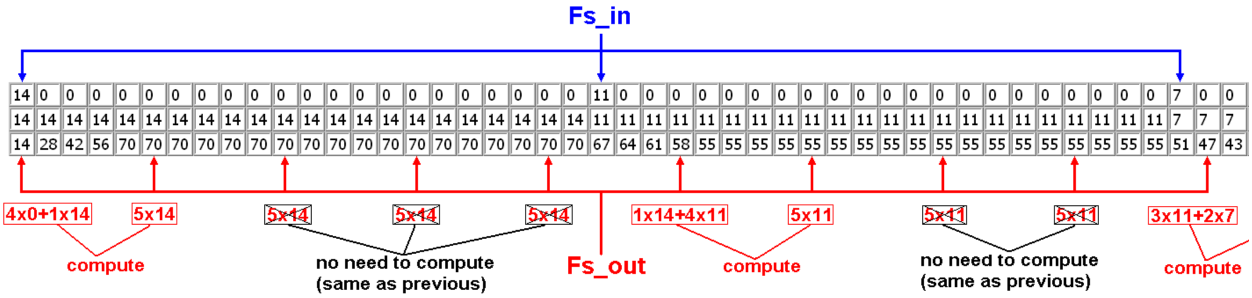
The formulas (2) and (3) can be used together to find locations and amplitudes of alias tones in the output spectrum of SRC.

The table containing the values for the sampling rate ratios shows that the lowest operating frequency for the SRC module using CIC filtering technique is 41.6MHz and the highest can go up to ~1.1GHz. The master clock is 13MHz. So, a standard way for implementing the CIC filters is impossible. Yet, some features of these filters can be exploited to allow in the end the use of an algorithm able to implement them, at a lower operating frequency. These key features are:

1. The up-sampling is a process of zero padding;
2. The transfer function of the module at the operating frequency is composed from two different boxcar FIR filters, with all the coefficients equal to 1;
3. The down-sampling is a process of throwing away many samples, therefore making the computation of those samples useless.

The problem reduces to finding the best digital architecture that will process the data in the same way an SRC operating at the required frequency will do it. The diagram below (Figure 10) describes what operations are performed inside the DAC_SRC module.

- The first row contains the upsampled input data,
- The second row contains data after the first boxcar FIR filter (U taps);
- The third row contains the data after the second boxcar FIR filter (D taps);
- In all of the cases the output sample is a function of the input sample and the value of D, excepting the case when a new input sample falls between two output samples;
- When a new input sample falls between two output samples the output is a function of the current and previous samples and two coefficients generated using a modulo counter procedure;
- The modulo counter provides information regarding the relative time-position of the input sample within the two output samples.



14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0	0
14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	7	7	7
14	28	42	56	70	70	70	70	70	70	70	70	70	70	70	70	70	70	70	70	70	70	67	64	61	58	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	51	47	43

Figure 10. Numerical-algorithm to implement the DAC-SRC (U=22, D=5).

As it was mentioned in [1], an interpolation will cause imaginary parts to show up in the spectrum, while the decimation will cause aliases. Since we are doing both in this case, one can expect to see many of these artifacts in the DAC-SRC output spectra. But the important aspect is to have all these unwanted tones far away from the audible band and with energies that are several orders of magnitude below the signal of interest.

Figure 11 and Figure 12 are capturing some of the most critical cases in terms of how clean the spectra will be after the rate conversion. They were placed here just to familiarize the reader with this idea. The following paragraphs will have more spectra related to this. Remembering that although the maximum audible frequency for the human ear is a theoretical value of 20 kHz it's advisable to keep the energy at higher frequency as low as possible. This will prevent the transistors from the amplification stage to behave in non-linear fashion, therefore causing distortion.

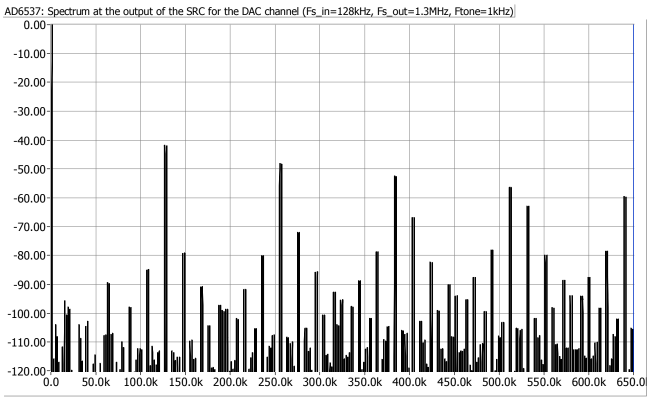


Figure 11. Output spectrum from DAC-SRC with input signal at 1 kHz.

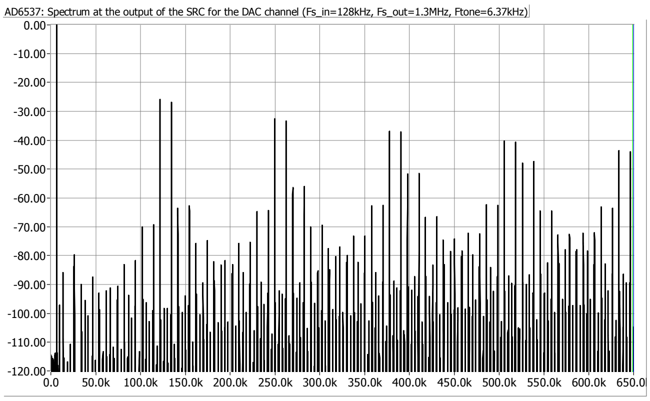


Figure 12. Output spectrum from DAC-SRC with input signal at 6.37 kHz.

II. SRC FOR THE DAC PATH: DETAILED DIGITAL HARDWARE IMPLEMENTATION

Step 1: We start from Figure 2 by writing the transfer functions of different blocks in the z domain. Figure 13 shows the resulted equivalent block diagram.

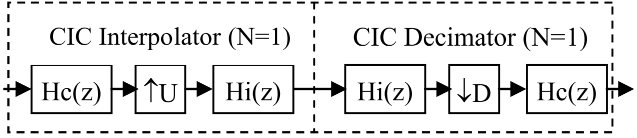


Figure 13. Two first order CIC filters concatenated.

$$Hi(z)=\left(\frac{1}{1-z^{-1}}\right);Hc(z)=\left(1-z^{-1}\right) \tag{5}$$

Step 2: The up-sampling block is pushed to the left side while the down-sampling block is pushed to the right side and the two transfer functions are merged into a single one (Figure 14). Noble identities 1 and 2 were applied during this process.

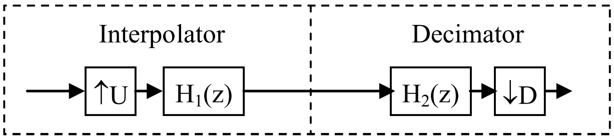


Figure 14. Equivalent diagram of concatenated CIC filters

$$H_1(z)=Hi(z)\cdot Hc(z^U)$$

$$H_2(z)=Hi(z)\cdot Hc(z^D)$$

$$H(z)=H_1(z)\cdot H_2(z)=\left(\frac{1-z^{-U}}{1-z^{-1}}\right)\cdot\left(\frac{1-z^{-D}}{1-z^{-1}}\right)\Leftrightarrow \tag{6}$$

$$H(z)=\left(\sum_{k=0}^{U-1}z^{-k}\right)\cdot\left(\sum_{k=0}^{D-1}z^{-k}\right)$$

Step 3: A series of equivalent transformations are performed on the transfer function to bring it into the general format of an FIR filter transfer function. The formula obtained for the coefficients c_k is valid only when U>D.

$$H(z)=\left(\sum_{k=0}^{U-1}z^{-k}\right)\cdot\left(\sum_{k=0}^{D-1}z^{-k}\right)=\sum_{k=0}^{U+D-2}c_k\cdot z^{-k}$$

$$with\ c_k=\begin{cases}k+1,\ if\ k<D\\ D,\ if\ D\le k<U\\ U+D-k-1,\ if\ k\ge U\end{cases}, \tag{7}$$

$$k=\overline{0,(U+D-2)}$$

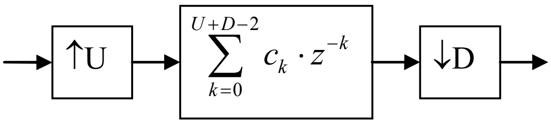


Figure 15. Starting multirate filtering diagram for the SRC-DAC.

Step 4: When dealing with multi-rate digital systems, the polyphase decomposition can lead to computationally efficient structures [Har04]. In this case, it was also a solution to overcome the problem of a very high clock frequency.

The polyphase decomposition will be performed by grouping together the terms of the form ck·z-k and ck+U·z-(k+U) of the transfer function. Since there are U+D-1 terms in the transfer function, we will add (U-D+1) extra coefficients from cU+D-1 to c2U-1 whose value will be equal to zero and will introduce more terms from z-(U+D-1) to z-(2U-1) in the transfer function, which will have now 2U terms. This will not affect the transfer function at all, but will allow grouping all the terms as planned and get U polyphase components (filters). In the end the new formula for H(z) becomes:

$$H(z)=\sum_{k=0}^{U-1}(c_k\cdot z^{-k}+c_{k+U}\cdot z^{-k-U})$$

$$\Updownarrow$$

$$H(z)=\sum_{k=0}^{U-1}(c_k+c_{k+U}\cdot z^{-U})\cdot z^{-k} \tag{8}$$

$$with\ c_k=\begin{cases}k+1, & if\ k<D\\ D, & if\ D\le k<U\\ U+D-k-1, & if\ U\le k<U+D-1\\ 0, & if\ k\ge U+D-1\end{cases}$$

$$k=\overline{0,(2\cdot U-1)}$$

By noting with Ek(zU) the resulted U polyphase components of the initial filter, a more compact way of writing the new formula for H(z) is obtained:

$$H(z)=\sum_{k=0}^{U-1}z^{-k}\cdot E_k(z^U) \tag{9}$$

$$E_k(z^U)=c_k+c_{k+U}\cdot z^{-U},\ with\ k=\overline{0,U-1}$$

Observation: one thing to notice at this stage and that will be used later, in the implementation phase, is that the pair of coefficients (c_k, c_{k+U}) can take values of the form either (k+1, U+D-k-1) or either (D, 0), depending on the value of k. This can be deducted by analyzing formula (9).

Figure 16 shows the corresponding polyphase decomposition of the full SRC architecture.

This new block diagram of the SRC module is still not able to solve the problems raised by the impossibility of having a clock at the required high frequency. Therefore, some more transformations will be applied to this architecture.

The first modification in the architecture involves moving the up-sampling block in each of the branches. Each filter from the branches will be preceded by an up-sampling unit as it can be seen in Figure 17.

Using Noble's identity [3], the up-sampling blocks are pushed more to the left side in front of the transfer functions $E_k(z^U)$. Figure 18 shows the new equivalent block diagram.

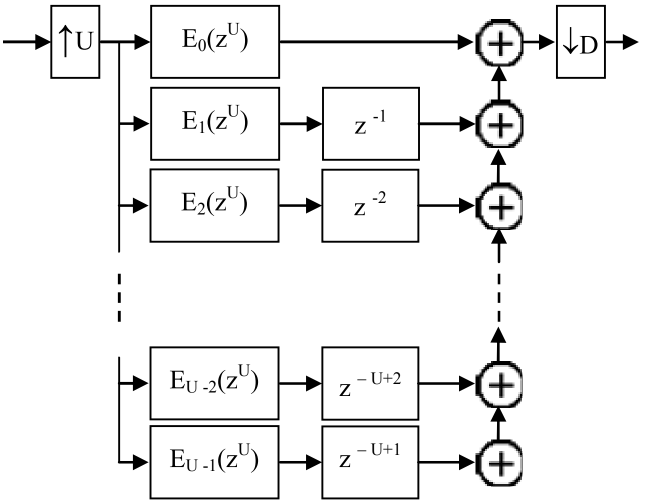


Figure 16. Polyphase decomposition of the filtering for DAC-SRC.

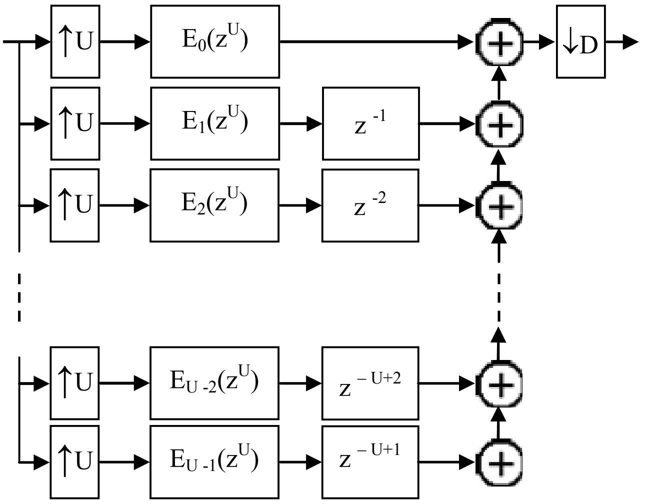


Figure 17. Equivalent diagram for the polyphase decomposition of DAC_SRC.

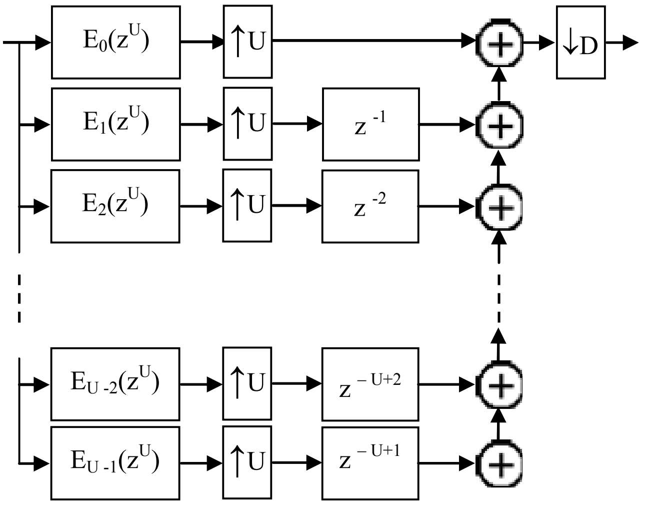


Figure 18. Equivalent diagram for the polyphase decomposition

It can be seen how the output of each branch filter is up-sampled and delayed, the delay being z^{-k} in the k-th branch. Finally, the outputs of the U branches are summed to form the final interpolated output signal. Due to zero-packing and different branch delays, only one branch filter output actually contributes to the final output at any given instant (while the other branch filter outputs are zero). Thus, the final structure can be implemented by multiplexing the branch filter outputs using a commutative switch.

Therefore, the back-end delays and the up-samplers can actually be discarded by simply picking the right k-th branch output at the right time using a commutative switch. The final structure is presented in Figure 19.

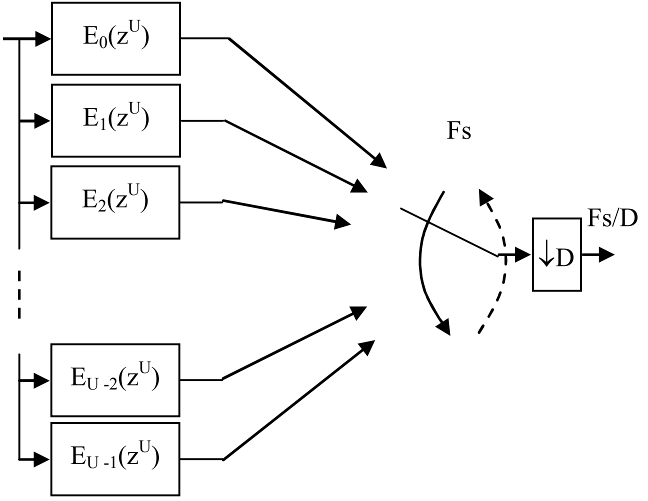


Figure 19. Final polyphase solution for DAC-SRC.

The order in which the commutative switch rotates during one cycle, should be consistent with the order in which the samples arrive to the SRC module and therefore it will start with first branch (according to the indexes used) going down to the last.

Figure 20 is exactly like the previous one, except the fact that the transfer functions of the branches were replaced with their equivalent formulas.

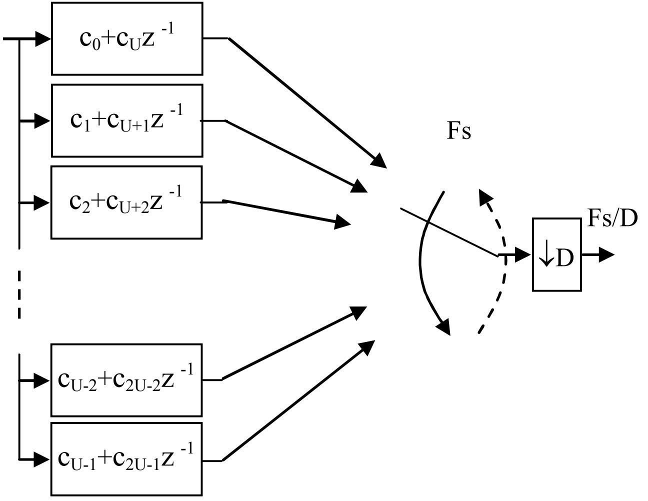


Figure 20. Final detailed polyphase solution for DAC-SRC.

Because the down-sampling block will skip D-1 samples it means we don't need to compute them and therefore the commutative switch can skip the branches that output those samples. This will allow it to work at the lower frequency (the one after the down-sampling block).

At this stage there are no more impediments in obtaining a digital implementation for the block diagram of the SRC module.

All the blocks required to implement the block diagram from Figure 20 are common standard blocks often used in the implementation of FIR filters (adders, multipliers, memory elements). The only one that is a bit different is the commutative switch and this one will be implemented using a modulo counter.

In Figure 21 is presented one solution for the digital implementation of the SRC module.

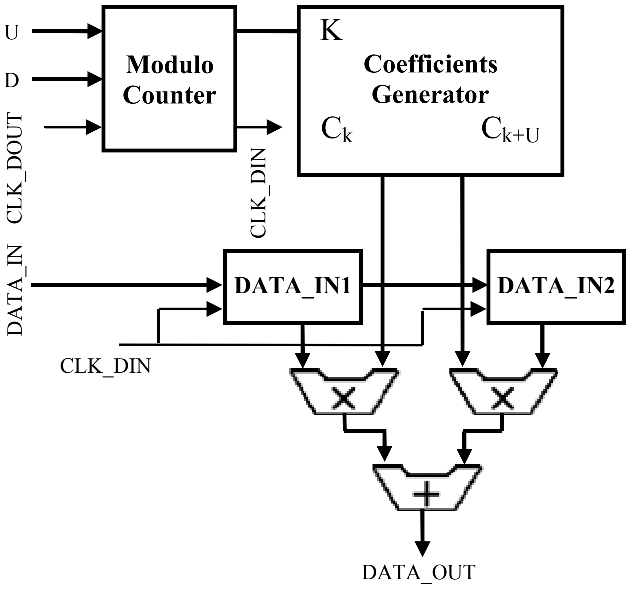


Figure 21. Digital hardware solution

The “Coefficients Generator” block will generate the required coefficients based on the input value K and eq. (9). By using the observation made in relation that equation, this block can be optimized since many pairs of coefficients have the value (D, 0). More, during one cycle of the commutative switch (or modulo counter) from the values 0 to U-1, there are several consecutive branch filters that use the pairs of coefficient with this value, while the input samples are also the same for all of these filters. This means that once we compute the output for the first of them we can simply use that value for the rest, therefore the number of computations required can be reduced.

All these advantages together with the availability of a faster clock (MCLK) were used during the digital implementation of this module.

The size of the accumulators (and DATA_OUT) depends on the size of DATA_IN and the values for D. The gain introduced by this block is equal to D and therefore the extra number of bits required for DATA_OUT (compared to DATA_IN) is $\log_2(D)$. The biggest value for D is 441 and this will require 9 extra bits. This will cover the other cases, when D has smaller values.

An optimized digital architecture for the DAC sample rate converter is shown in Figure 22.

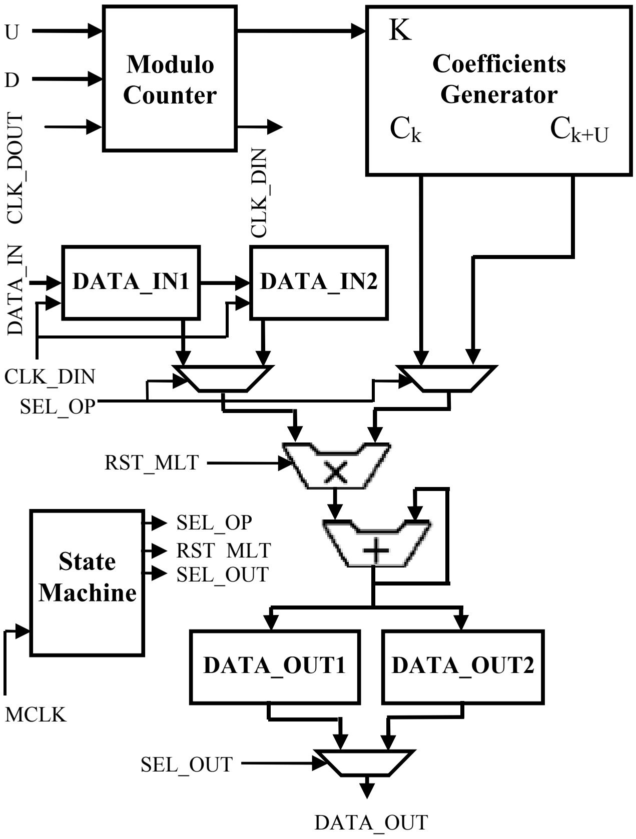


Figure 22. Optimized digital hardware solution.

The multiplier is implemented as a bit-serial multiplier and is controlled by the state machine. The various multiplexers are also controlled by the state machine. Because the multiplier is serial, two registers for keeping the computed output values and send them out at the right time had to be used.

The waveforms from the plot shown below are obtained by simulating the RTL code used in the digital implementation of the DAC sample rate converter.

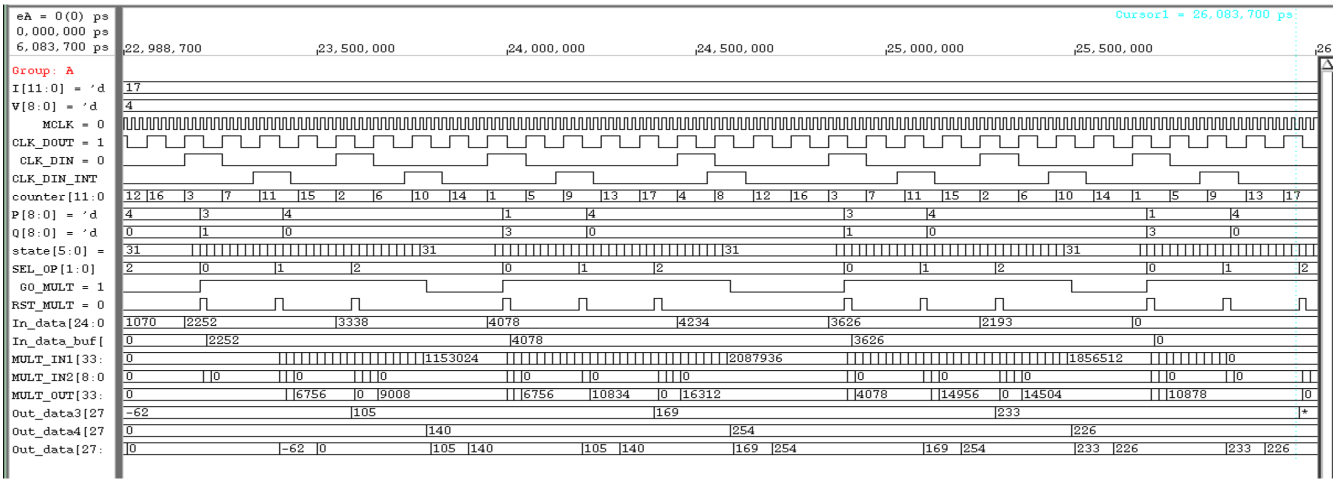


Figure 23. Digital waveforms from the RTL simulation.

Smaller values than the real ones were chosen for the up-sampling and down-sampling blocks (17 and 4, as opposed to 650 and 96). Variables P and Q are equivalent to ck and ck+D, whereas Out_data1 and Out_data2 are equivalent to Acc1 and Acc2.

Signals SEL_OP is used to select what operands are going to be used in the bit-serial multiplier and GO_MULT is used to control the multiplier.

In the zoomed version of the waveforms (Figure 24) can be seen more details on how the state machine is controlling the serial multiplier and other blocks in the circuit:

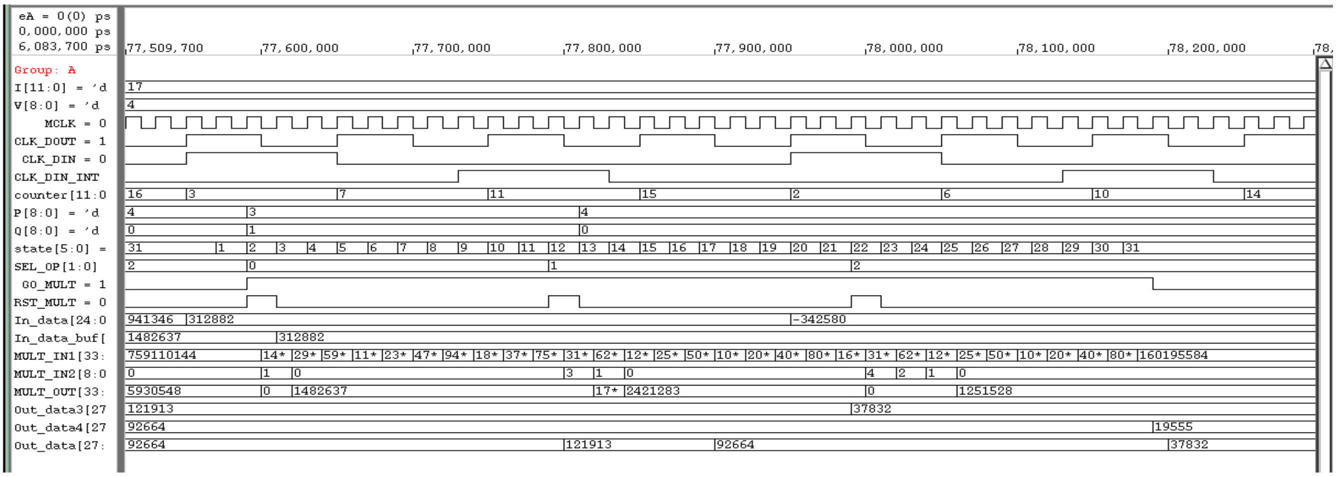


Figure 24. Zoomed digital waveforms from the RTL simulation.

The performance of this circuit can be analyzed from the next waveforms (Figure 23 - Figure 33) showing the spectrum of the signals at the input and output of the DAC-SRC module for various sampling rate conversions (related to different audio standards) and using various input tones from the audio bandwidth.

The chosen tones are closer to the higher frequency end of the audio bandwidth. The reason for the choice comes from the fact that the nulls of the anti-imaging and anti-aliasing filtering used inside the DAC-SRC attenuate better frequencies that are closer to DC. Therefore, in order to show the performance of this solution, I wanted capture the most critical situations. The other ones will obviously have better performance.

As a general observation, one can see that the input spectra have some spurious tones from the previous interpolation stages of the signal path. These are usually at 80dB or 90dB below the main tone, which is ok. In the output spectra, there are much more spurious tones created by the DAC-SRC, but the request is to have them attenuated by 20-30dB below the tone of interest. There will be some more low-pass filtering applied to the signal by the digital delta-sigma modulator which will push these spurious tones even lower.

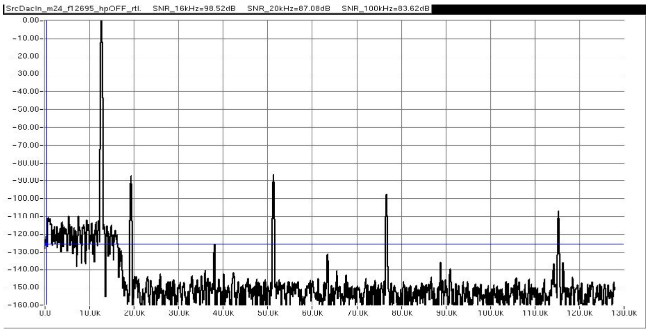


Figure 25. Input spectrum with 12.7 kHz as main tone (32 kHz audio standard): Fs_in = 8*32 kHz, Fs_out= 2.6MHz, Tone_in = ~12.7 kHz. Input of DAC_SRC.

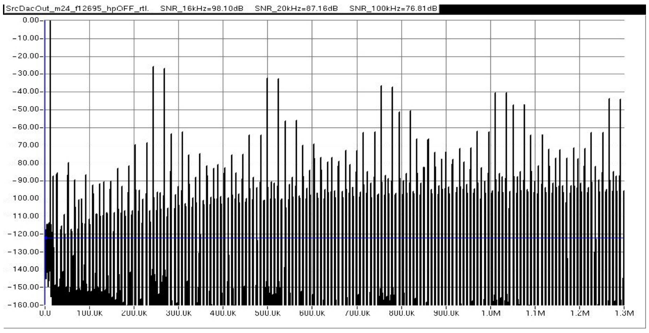


Figure 26. Output spectrum with 12.7 kHz as main tone (32 kHz audio standard) Fs_in = 8*32 kHz, Fs_out= 2.6MHz, Tone_in = ~12.7 kHz. Output of DAC_SRC (full spectrum).

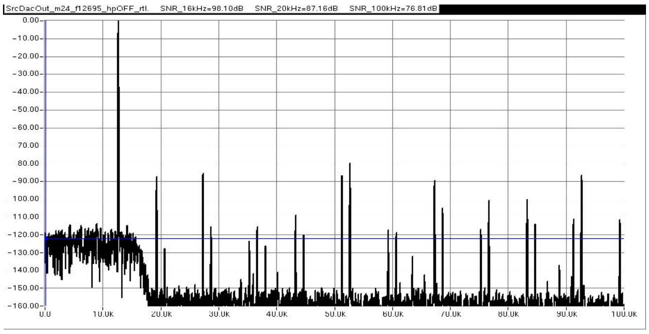


Figure 27. Zoomed output spectrum with 12.7 kHz as main tone (32 kHz audio standard): Fs_in = 8*32 kHz, Fs_out= 2.6MHz, Tone_in = ~12.7 kHz. Output of DAC_SRC.

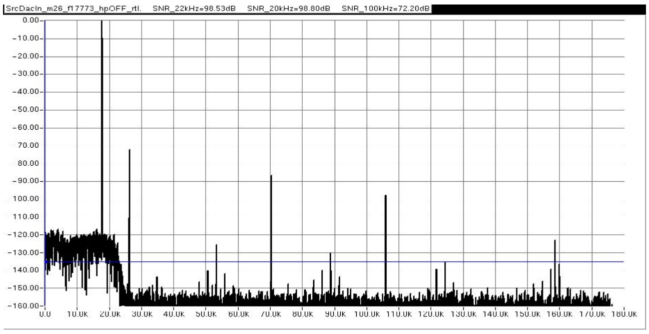


Figure 28. Input spectrum with 17.8 kHz as main tone (44.1 kHz audio standard): Fs_in = 8*44.1 kHz, Fs_out= 2.6MHz, Tone_in = ~17.8 kHz. Input of DAC_SRC.

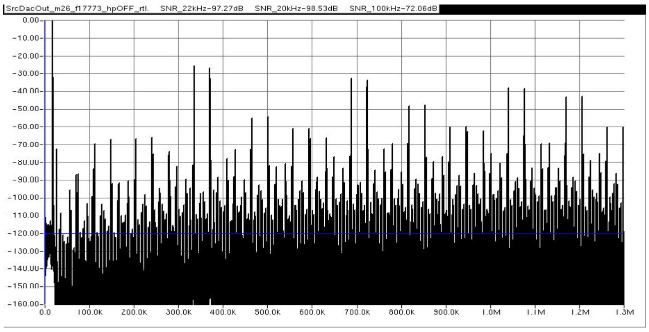


Figure 29. Output spectrum with 17.8 kHz as main tone (44.1 kHz audio standard): Fs_in = 8*44.1 kHz, Fs_out= 2.6MHz, Tone_in = ~17.8 kHz. Output of DAC_SRC (full spectrum).

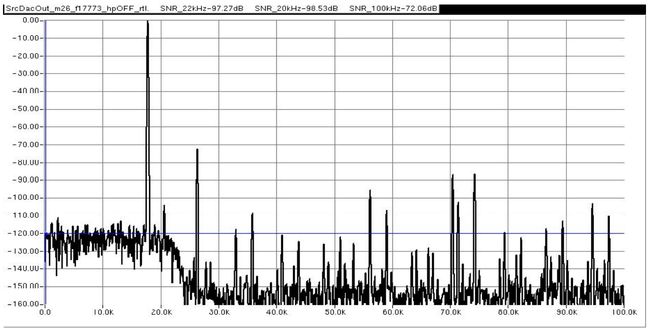


Figure 30. Zoomed output spectrum with 17.8 kHz as main tone (44.1 kHz audio standard): Fs_in = 8*44.1 kHz, Fs_out= 2.6MHz, Tone_in = ~17.8 kHz. Output of DAC_SRC.

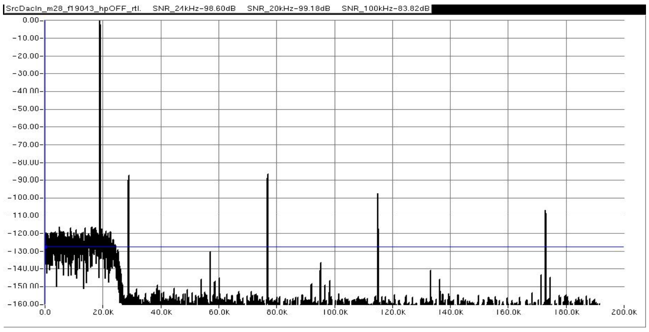


Figure 31. Input spectrum with 19 kHz as main tone (48 kHz audio standard): Fs_in = 8*48 kHz, Fs_out= 2.6MHz, Tone_in = ~19 kHz. Input of DAC_SRC.

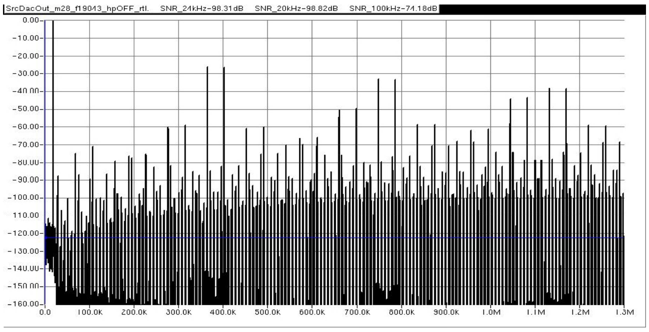


Figure 32. Output spectrum with 19 kHz as main tone (48 kHz audio standard): Fs_in = 8*48 kHz, Fs_out= 2.6MHz, Tone_in = ~19 kHz. Output of DAC_SRC (full spectrum).

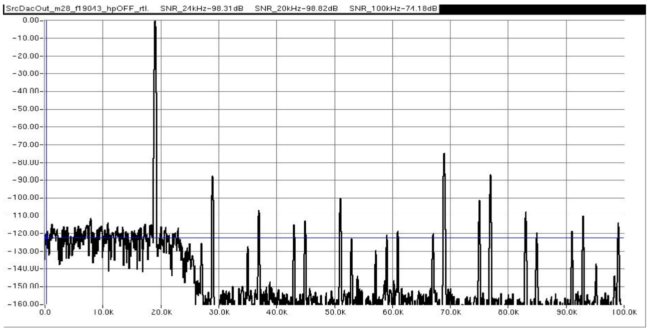


Figure 33. Zoomed output spectrum with 19 kHz as main tone (48 kHz audio standard): Fs_in = 8*48 kHz, Fs_out= 2.6MHz, Tone_in = ~19 kHz. Output of DAC_SRC.

III. CONCLUSIONS

This paper shows the practical implementation that was done to solve a real demand from the mobile communications devices. The method that was applied is a novel one and it can be extended to allow more advanced filtering if some other applications will require. The filtering branches of the polyphase decomposition can be changed to have a different transform function, if a certain user requires some other type of filter.

While being elegant and flexible, this method is also quite efficient in terms of the digital resources (logic gates) that are required.

Until now, most of the existing delta-sigma DACs were considering the usage of only integer values for interpolation and decimation. The solution presented here can open the door for the usage of a fractional value and this can be an excellent solution if the available clock of the system is not an integer multiple of the sampling frequency for the DACs.

Even if this work was applied to a specific audio problem concerning sampling rates, the applications in which ideas presented here can be applied are numerous and growing.

There is a patent application that was issued in 2007 and which protects the ideas presented in this document [4].

REFERENCES

[1] L. Bening, H. Foster, Principles of Verifiable RTL Design, 2nd edition, Kluwer Academic Publishers Norwell, MA, USA, 2001.

[2] G. Antonesei, C. Turcu, A. Graur, Basic Consideration for Signal Processing Solutions Used in Sigma-delta Based ADC and DAC Converters, Advances in Electrical and Computer Engineering, ISSN: 1582-7445, e-ISSN: 1844-7600,doi: 10.4316/aece, vol. 10, number 1, 2010, available at www.aece.ro.

[3] P. Schniter, Noble Identities, available at http://cnx.org/content/m10432/2.12/

[4] G. Antonesei, Sample Rate Converter For Reducing The Sampling Frequency Of A Signal By A Fractional Number, US Patent 7236110, 2007.