International Journal of Electronics
Vol. 97, No. 6, June 2010, 663–676

Taylor & Francis
Taylor & Francis Group

Fully synthesised decimation filter for delta-sigma A/D converters

Hyungdong Roh, Sanho Byun, Youngkil Choi, and Jeongjin Roh*

The Department of Electrical Engineering, Hanyang University, Ansan 426-791, Korea

(*Received 21 January 2009; final version received 27 January 2010*)

Digital decimation filters are used in delta-sigma analogue-to-digital converters to reduce the oversampled data rate to the final Nyquist rate. This paper presents the design and implementation of a fully synthesised digital decimation filter that provides a time-to-market advantage. The filter consists of a cascaded integrator-comb filter and two cascaded half-band FIR filters. A canonical signed-digit representation of the filter coefficients is used to minimise the area and to reduce the hardware complexity of the multiplication arithmetic. Coefficient multiplications are implemented by using shifters and adders. This three-stage decimation filter is fabricated by using 0.25-μm CMOS technology with an active area of 1.36 mm^2 and shows 4.4 mW power consumption at a clock rate of 2.8224 MHz. Experimental results show that this digital decimation filter is suitable for use in oversampled data converters and can be applied to new processes requiring a fast redesign time. This is possible because the filter does not have process-dependent ROM or RAM circuits.

Keywords: decimation filter; A/D converter

1. Introduction

Oversampled delta-sigma analogue-to-digital converters (ADCs) have been widely used in high-resolution data converters (Schreier and Temes 2005; Roh et al. 2008, 2009). Historically, reducing the silicon area and the power consumption of delta-sigma ADCs have been key design issues. The two main blocks of a delta-sigma converter are the analogue modulator and the digital decimation filter. The analogue input to the ADC is converted into oversampled digital data, which is then filtered and decimated by decimation filters (Gao and Tenhunen 1999; Dolecek and Mitra 2005; Laddomada 2008; Dolecek and Mitra 2008). The area and the power used by oversampled delta-sigma converters are dictated largely by the digital decimation filters, and ROM and RAM circuits are essential to efficient designs (Brandt and Wooley 1994). However, in deep sub-micron processes, transistor-scaling theory makes possible digital circuits that are significantly reduced in both area and power consumption. In addition, logic-synthesised field-programmable devices are commonly used for system verification by using hardware description languages (HDLs) (Meyer-Baese 2004). This new trend requires fully synthesizable digital blocks for rapid design and verification.

*Corresponding author. Email: jroh@hanyang.ac.kr

ISSN 0020-7217 print/ISSN 1362-3060 online
© 2010 Taylor & Francis
DOI: 10.1080/00207211003697830
http://www.informaworld.com

Conventional decimation filters implement memory elements by using process-dependent ROM and RAM circuits. ROM and RAM can be implemented within a small silicon, but they prevent a rapid design turn-around time when a new design is required. Recent decimation filter designs utilise field-programmable devices to reduce the time to market. We have used the Verilog HDL for the complete design and synthesis reported in this paper. The use of HDL provides time-to-market advantages due to the two approaches taken in system implementation. First, a new system can be implemented based on field programmable gate array (FPGA) (Yang and Fathy 2009). As the fabrication of a silicon chip takes more than one month, the FGPA-based hardware can enter the market at an earlier stage. Also, the HDL design and FPGA implementation give a high level of confidence in design verification. A failed chip design results in the loss of development time and fabrication expenses, and FPGA verification allows the design to be evaluated at an earlier time during fabrication. Second, the digital design, which includes ROM and RAM blocks, requires extra work in silicon design. The synthesised logic blocks should be interfaced with the ROM and RAM circuits that are generated by using dedicated CAD tools. Fully synthesised logic simplifies the design steps and reduces the design time.

This paper presents the details of the implementation of a fully synthesised digital decimation filter. The cascaded integrator-comb (CIC) filter and two half-band FIR filters using a full Verilog-HDL design enable designers to quickly implement filters suitable to the specifications of delta-sigma modulators and suitable to various CMOS technologies using HDL synthesis. This is done instead of using RAM for data transfer and ROM for coefficient storage. The order and coefficients of our decimation filter are determined by using MATLAB simulations. This paper is organised as follows. In section 2, the architecture and characteristics of the decimation filter are presented. Hardware implementation is also described. In section 3, the physical design and experimental results are presented. Conclusions are provided in section 4.

2. Filter design and implementation

Figure 1 shows the designed digital decimation filter architecture. We chose a three-stage filter to minimise the size and power consumption of the decimation filter. The fifth-order CIC filter reduces the sampling rate by a factor of 16 and two cascaded half-band FIR filters reduce the remaining sampling rate to the Nyquist rate of 44.1 kHz. As the CIC filter has a passband droop, a relatively simple extra stage might be required as a droop-compensation filter (Schreier and Temes 2005). In this paper, however, only three stages are studied to evaluate the hardware implementation.

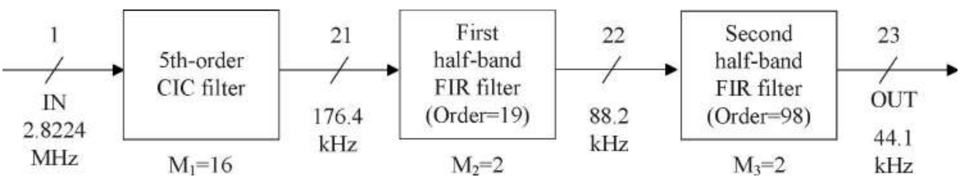


Figure 1. Three-stage digital decimation filter.

We use the Hogenauer CIC filter as the first stage to reduce the sampling rate (Hogenauer 1981). The order of the CIC filter must be greater than the noise-shaping order of the delta-sigma modulator (Candy 1986). As the noise-shaping order of the delta-sigma modulator is assumed to be 4 in our design, a fifth-order CIC filter is selected for the first stage, as illustrated in Figure 2. In the design of the fifth-order CIC filter, because the feedback coefficient of each integrator is unity, overflow problems can occur.

To solve overflow problems, the fifth-order CIC filter is implemented by using a two's complement format (Hogenauer 1981).

$$Word\ length = L(\log_2 RM) + B_{in}, \tag{1}$$

where L is the stages of the CIC filter, M is the number of delays in the comb section, and R is the down-sampling factor. The fifth-order CIC filter with a sign-extended input-data width (B_{in}) of 2 bits (01 or 11), $L = 5$, $R = 16$, and $M = 2$, requires a word length as shown guarantee no overflow errors (Meyer-Baese 2004). In this paper, the required word length of 27 bits was calculated by using Equation (1). As shown in Figure 1, the final output width of the digital decimation filter is 23 bits, and the bit width is increased at the two half-band filter stages by 1 bit each following the CIC filter. In the fifth-order CIC filter, we drop the lower 6 bits of the output of the comb section so that the final output width of the fifth-order CIC filter becomes a 21-bit word. Down-sampling, used elsewhere, by a factor of 16 is implemented by using operating registers in the integrating section at 2.8224 MHz and registers in the comb section at 176.4 kHz. An MUX (multiplexer) is used to change the modulator output (1 bit) into a two's complement format. Then a sign-extension operation, by which the two's complement output of the delta-sigma modulator is widened by 25 extension bits, precedes the integration operation. The subtraction operation for the two's complement words is implemented by using inverters and a carry input. Figure 3 shows the overall magnitude response of the designed CIC filter.

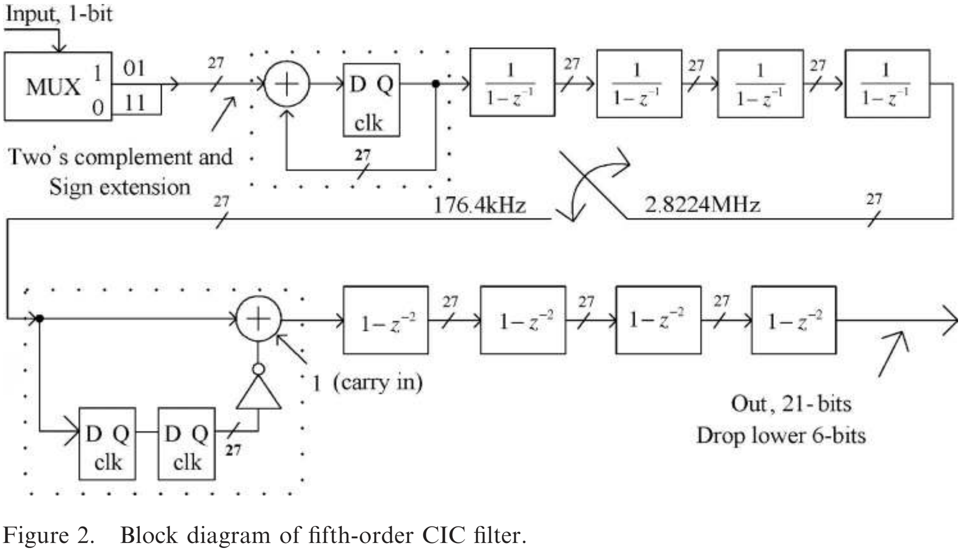


Figure 2. Block diagram of fifth-order CIC filter.

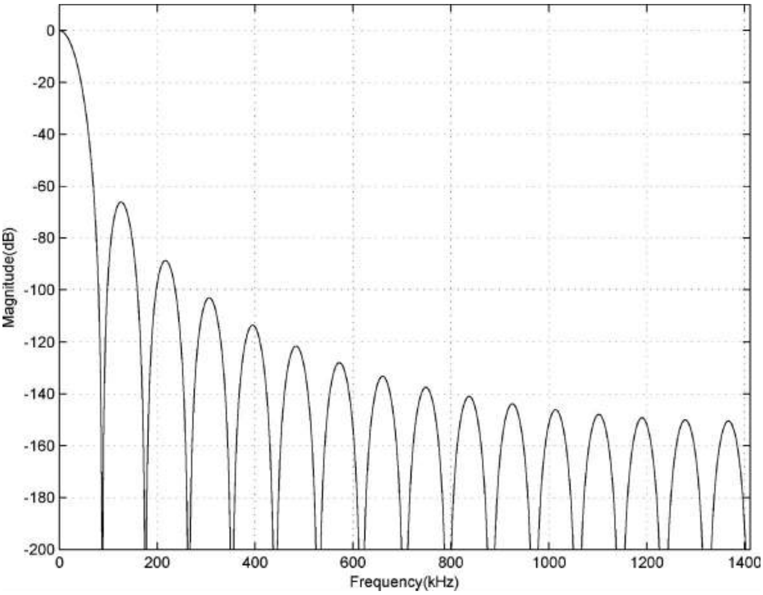


Figure 3. Overall magnitude response of the three-stage CIC filter.

The fifth-order CIC filter has a simple structure that can largely reduce the sampling rate. However, minimum attenuation at the stopband is insufficient. Therefore we use a cascade of two half-band FIR filters to increase attenuation at the stopband. In this paper, the half-band FIR filter is implemented with a polyphase direct-form filter to reduce hardware complexity and power consumption (Proakis and Manolakis 1996). The output sampling rate (176.4 kHz) of the fifth-order CIC filter is the same as the input sampling rate of the first half-band FIR filter. The output sampling rate of the first half-band FIR filter is 88.2 kHz because of the down-sampling of the half-band decimation filter. The normalised passband ($F_{N,pass}$) and stopband ($F_{N,stop}$) in the first half-band FIR filter are

$$F_{N,pass} = \frac{20}{88.2} = 0.227, \quad F_{N,stop} = \frac{68.2}{88.2} = 0.773. \tag{2}$$

In Equation (2), the first half-band FIR filter can be implemented by using a low-order filter because the transition band is relatively wide with $F_{N,stop} - F_{N,pass} = 0.546$. The first half-band FIR filter is implemented by using Goodman and Carey's F9 half-band FIR filter (Goodman and Carey 1977).

Figure 4 shows the magnitude response of the first half-band filter. The 19th-order F9 filter has a minimum attenuation of 75 dB at the stopband and a low passband ripple of 0.0015 dB. Therefore the filter is suitable as the first half-band FIR filter in this design. The total number of non-zero coefficients in the Goodman and Carey F9 filter is six, but the center of the odd coefficient can be implemented by shifting to the right by 1 bit. Therefore storage for only five coefficients is needed for multiplication. In this paper, multiplication is implemented by means of a shifter and an adder to reduce area and hardware complexity without the use of complicated multipliers.

It is essential to convert the coefficients of the half-band FIR filters into digital codes for hardware implementation. The common digital code, such as two's complementary or binary coded decimal (BCD), are easy to undestand, but the hardware overhead will be increased. We have compared the the two's complement and CSD format for multiplication (Hewlitt and Swartzlander 2000) below. The number of shift-and-add operations increases as the number of non-zero coefficient

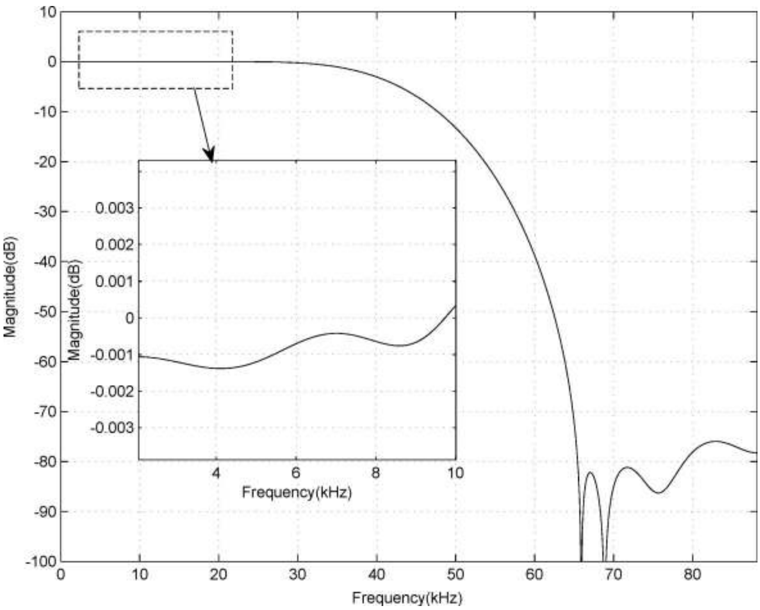


Figure 4. Magnitude response of the F9 filter.

Table 1. Two's complement number of the first half-band FIR filter.

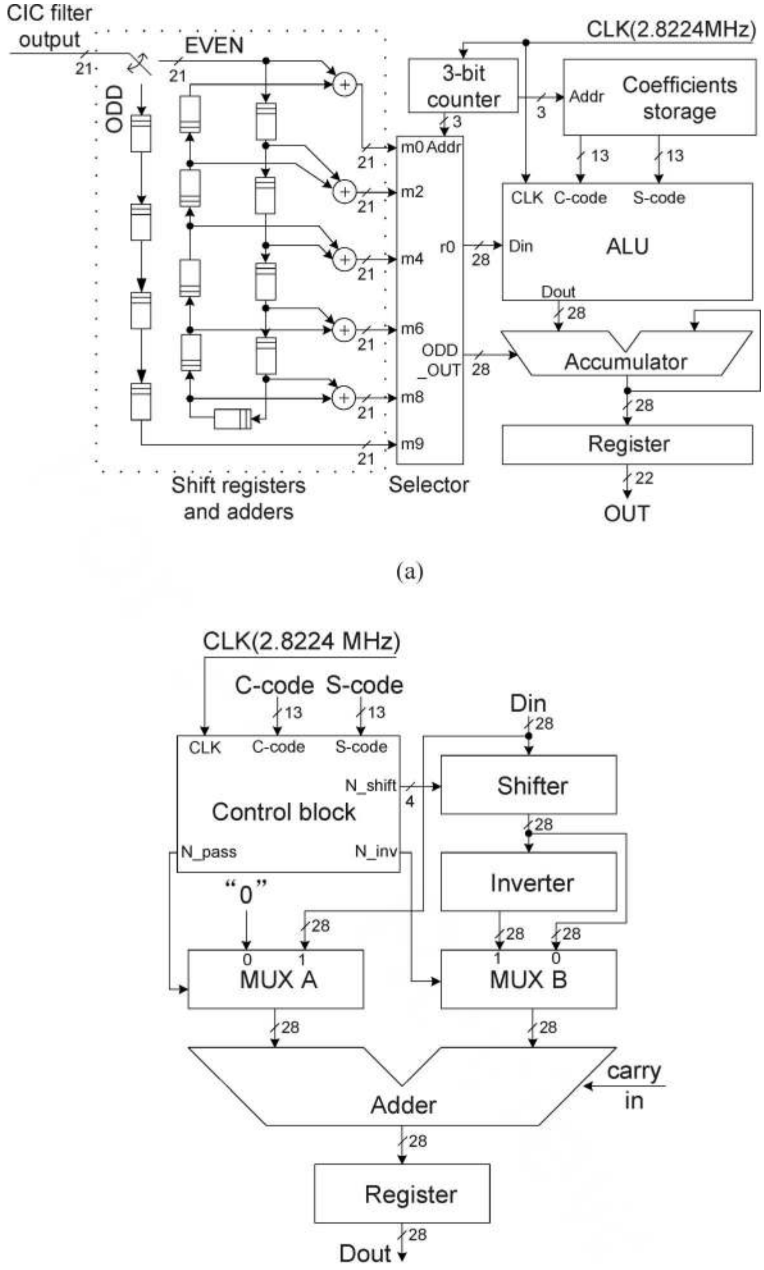
Coefficient	Two's complement format													
18	0	0	0	0	0	0	0	0	0	1	0	0	1	0
−116	1	1	1	1	1	1	1	0	0	0	1	1	0	0
429	0	0	0	0	0	1	1	0	1	0	1	1	0	1
−1278	1	1	1	0	1	1	0	0	0	0	0	0	1	0
5042	0	1	0	0	1	1	1	0	1	1	0	0	1	0

Table 2. CSD number of the first half-band FIR filter.

Coefficient	CSD representation format													
18	0	0	0	0	0	0	0	0	0	1	0	0	1	0
−116	0	0	0	0	0	0	−1	0	0	1	0	−1	0	0
429	0	0	0	0	1	0	0	−1	0	−1	0	−1	0	1
−1278	0	0	0	0	−1	0	−1	0	0	0	0	0	1	0
5042	0	1	0	1	0	0	0	−1	0	−1	0	0	1	0

values increases. In this paper, the CSD representation is used to reduce the number of non-zero values.

In CSD representation, each digit can be one of three values [1, 0, −1]. Table 1 shows the coefficients for the first half-band filter represented in two's complement



(a)

(b)

Figure 5. (a) The structure of the F9 filter. (b) The structure of the ALU.

format and Table 2 shows the same coefficients represented in CSD format. When making a comparison between Table 1 and Table 2, we can see that the CSD representation provides fewer non-zero values than does the two's complement representation.

Figure 5(a) illustrates the structure of the first half-band FIR filter. The output sequence of the fifth-order CIC filter is separately stored in odd or even shift registers by using an alternating switch. It cycles through shift registers and then is added to the delayed output of the shift register for the even branch. Table 3 shows that the coefficients can be represented by the addition of 2^{-n} for multiplication by using a right-shift operation of the data. C-code and S-code storage are used to distinguish between a -1 digit and a 1 digit in the CSD representation of coefficients. The C-code and S-code are stored by using Verilog-HDL synthesis and are transferred to

Table 3. The storage of coefficients represented by CSD format.

Coefficient	$429/2^{14} = 2^{-5} - 2^{-8} - 2^{-10} - 2^{-12} - 2^{-14}$													
CSD code	0	0	0	0	1	0	0	-1	0	-1	0	-1	0	1
C-code	0	0	0	0	1	0	0	1	0	1	0	1	0	1
S-code	0	0	0	0	0	0	0	1	0	1	0	1	0	0

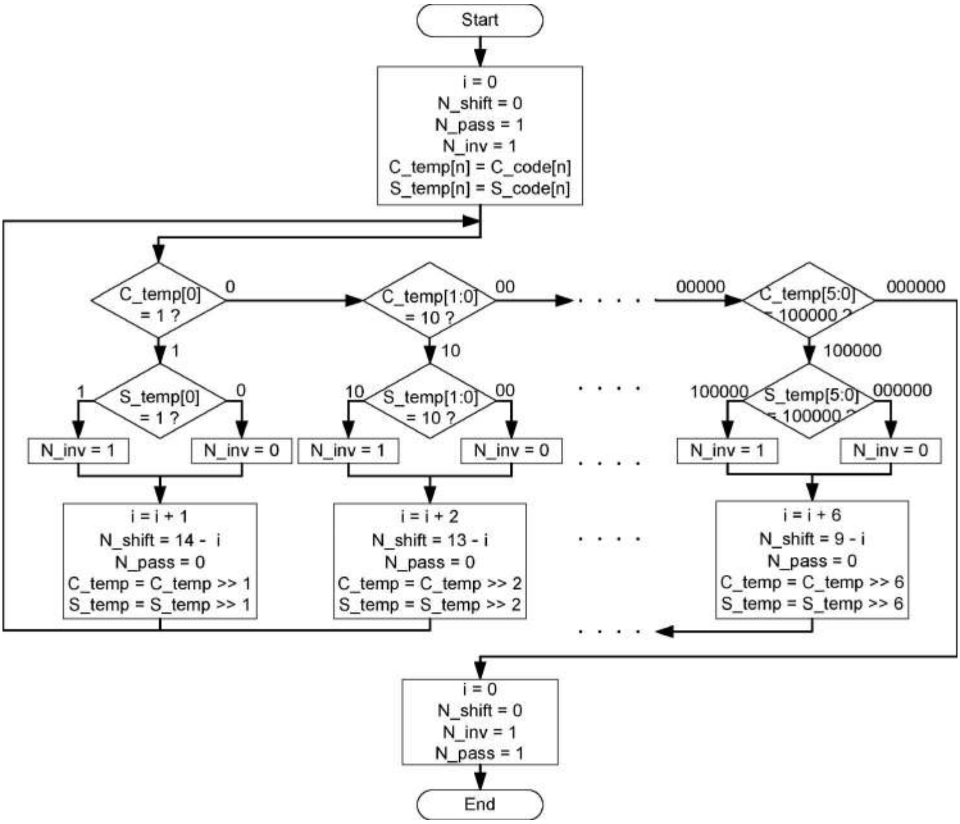
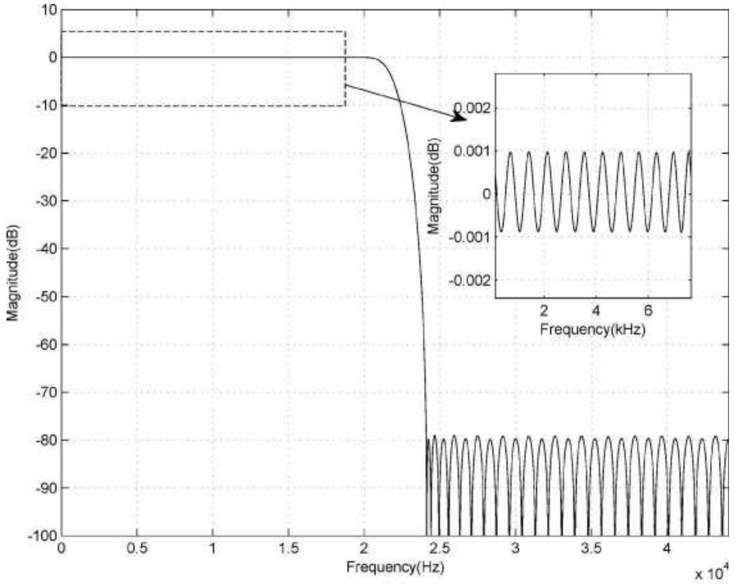


Figure 6. Flow chart of the control block.



(a)

$h(0) \;\; = h(98) = \;\; 1.0657310686E\text{-}04 \;\; = 2^{-13} - 2^{-16} - 2^{-22}$

$h(2) \;\; = h(96) = -1.4615058899E\text{-}04 \;\; = 2^{-13} - 2^{-15} + 2^{-17} - 2^{-20} - 2^{-22}$

$h(4) \;\; = h(94) = \;\; 2.4271011353E\text{-}04 \;\; = 2^{-12} - 2^{-19} + 2^{-22}$

$h(6) \;\; = h(92) = -3.7789344788E\text{-}04 \;\; = -2^{-11} + 2^{-13} - 2^{-16} + 2^{-18} + 2^{-22}$

$h(8) \;\; = h(90) = \;\; 5.6076049805E\text{-}04 \;\; = 2^{-11} + 2^{-14} + 2^{-16} - 2^{-18} - 2^{-22}$

$h(10) = h(88) = -8.0275535583E\text{-}04 \;\; = -2^{-10} +2^{-12} - 2^{-14} - 2^{-17} - 2^{-20} - 2^{-22}$

$h(12) = h(86) = \;\; 1.11556053162E\text{-}03 \;\; = 2^{-10} + 2^{-13} + 2^{-16} + 2^{-19} - 2^{-21}$

$h(14) = h(84) = -1.51348114014E\text{-}03 \;\; = -2^{-9} + 2^{-11} - 2^{-14} + 2^{-16} - 2^{-19} - 2^{-21}$

$h(16) = h(82) = \;\; 2.01201438904E\text{-}03 \;\; = 2^{-9} + 2^{-14} - 2^{-18} + 2^{-20} + 2^{-22}$

$h(18) = h(80) = -2.62904167175E\text{-}03 \;\; = -2^{-8} + 2^{-10} + 2^{-12} + 2^{-14} - 2^{-18} - 2^{-22}$

$h(20) = h(78) = \;\; 3.38506698608E\text{-}03 \;\; = 2^{-8} - 2^{-11} - 2^{-15} - 2^{-18} - 2^{-20}$

$h(22) = h(76) = -4.30488586426E\text{-}03 \;\; = 2^{-8} - 2^{-11} + 2^{-13} - 2^{-15} - 2^{-19} + 2^{-21}$

$h(24) = h(74) = \;\; 5.41782379150E\text{-}03 \;\; = 2^{-7} - 2^{-9} - 2^{-11} + 2^{-14} - 2^{-16} + 2^{-21}$

$h(26) = h(72) = -6.76155090332E\text{-}03 \;\; = -2^{-7} + 2^{-10} + 2^{-14} + 2^{-16} - 2^{-19} + 2^{-21}$

$h(28) = h(70) = \;\; 8.38518142700E\text{-}03 \;\; = 2^{-7} + 2^{-11} + 2^{-13} - 2^{-15} - 2^{-17}$

$h(30) = h(68) = -1.035642623901E\text{-}02 \;\; = -2^{-7} - 2^{-9} - 2^{-11} - 2^{-13} + 2^{-16} + 2^{-18} + 2^{-20}$

$h(32) = h(66) = \;\; 1.277279853821E\text{-}02 \;\; = 2^{-6} - 2^{-8} + 2^{-10} + 2^{-14} +2^{-16} + 2^{-20} - 2^{-22}$

$h(34) = h(64) = -1.578283309937E\text{-}02 \;\; = -2^{-6} - 2^{-13} - 2^{-15} - 2^{-18} - 2^{-20}$

$h(36) = h(62) = \;\; 1.962518692017E\text{-}02 \;\; = 2^{-6} + 2^{-8} + 2^{-13} - 2^{-15} + 2^{-19}$

$h(38) = h(60) = -2.471423149109E\text{-}02 \;\; = -2^{-5} + 2^{-7} - 2^{-10} - 2^{-12} - 2^{-14} + 2^{-17} - 2^{-19} - 2^{-22}$

$h(40) = h(58) = \;\; 3.183484077454E\text{-}02 \;\; = 2^{-5} + 2^{-11} + 2^{-13} - 2^{-15} +2^{-18} + 2^{-20}$

$h(42) = h(56) = -4.267573356628E\text{-}02 \;\; = -2^{-4} + 2^{-6} + 2^{-8} + 2^{-12} + 2^{-14} - 2^{-16} + 2^{-18} - 2^{-21}$

$h(44) = h(54) = \;\; 6.163692474365E\text{-}02 \;\; = 2^{-4} - 2^{-10} + 2^{-13} - 2^{-17} - 2^{-20} - 2^{-22}$

$h(46) = h(52) = -1.0487747192383E\text{-}01 = -2^{-3} + 2^{-6} + 2^{-8} + 2^{-11} + 2^{-13} - 2^{-16} - 2^{-18} + 2^{-22}$

$h(48) = h(50) = \;\; 3.1789922714233E\text{-}01 = 2^{-2} + 2^{-4} - 2^{-9} - 2^{-11} + 2^{-15} - 2^{-19} - 2^{-21}$

$h(49) = \;\; 0.50000429153442 \;\; = 2^{-1}$

(b)

Figure 7. (a) Magnitude response of the second half-band FIR filter. (b) The filter coefficients.

the ALU according to the address created by the 3-bit counter. Figure 5(b) illustrates the structure of the ALU. The control block of the ALU generates N_shift, N_inv, and N_pass signals.

Figure 6 shows the flow chart for the control block. First, the control block finds the first 1 value from the C-code and compares it with the determined patterns. Then it checks the 1 value in the S-code. If the control block finds a value of 1 in the S-code then it sets the N _ inv signal to 1. This process effectively executes a -1 digit by using the CSD representation of the coefficients. The N _ shift signal represents the value n of 2^{-n}, as shown in Table 3, generated from the value i (from Figure 6), which points to the location of a 1 value in the C-code. The control block repeatedly performs the aforementioned steps until it finds the final 1 value in the C-code. The shifter shifts the input data to the right with an N _ shift signal value. If the N_inv signal has a value of 1, the MUX B passes the inverted shifter output. The first half-band FIR filter can use 32 clock cycles (2.8224 MHz/88.2 kHz) for multiplication operations. Hence, multiplication is implemented by using one ALU in the first half-band FIR filter. The implementation of the third stage is similar to that of the second stage. However, there are some differences between the first and the second half-band FIR filters. The order of the second half-band FIR filter is greater than that of the first half-band FIR filter, as the second half-band FIR filter has a significantly narrower transition band. It also has more restricted clock cycles to carry out coefficient multiplication. The input sampling rate of the second half-band FIR filter is 88.2 kHz and the output rate 44.1 kHz. The normalised passband and stopband of the second half-band FIR filter are

$$F_{N,pass} = \frac{20}{44.1} = 0.453, \quad F_{N,stop} = \frac{24.1}{44.1} = 0.546, \tag{3}$$

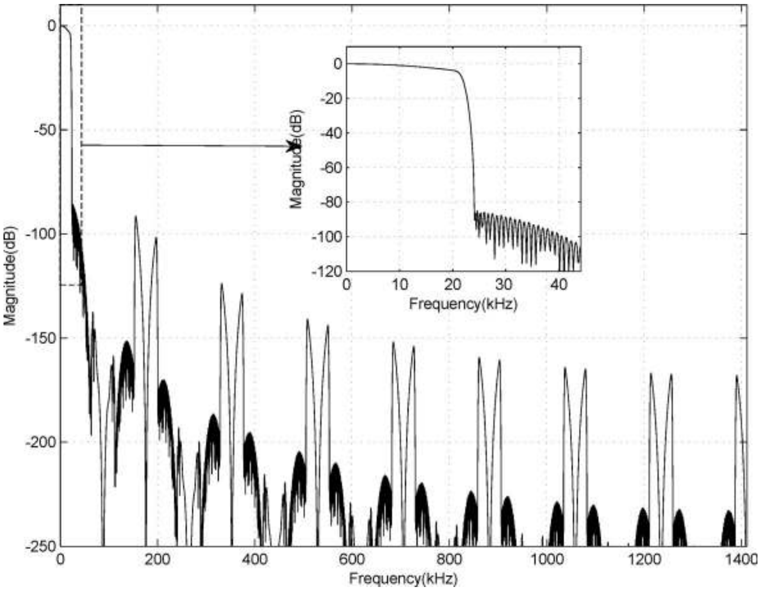


Figure 8. Magnitude response of decimation filter.

where the passband is 20 kHz and the stopband 24.1 kHz. Coefficients are selected for the second half-band FIR filter by using a MATLAB simulation employing a Remez exchange algorithm (Harris 2004).

The magnitude response of the second half-band FIR filter and the coefficients are shown in Figure 7. The second half-band FIR filter has a maximum of eight non-zero values for each coefficient and a total of 25 coefficients. These characteristics determine the need for a minimum of 200 (8 × 25) clock cycles for multiplication of all coefficients.

However, the second half-band FIR filter can use only 64 clock cycles (2.8224 MHz/44.1 kHz) for the multiplication operations. To solve this problem, we use five parallel ALUs in the second half-band FIR filter.

Figure 8 shows the overall magnitude response of the three-stage decimation filter. The passband is 0–20 kHz and the minimum stopband attenuation is larger

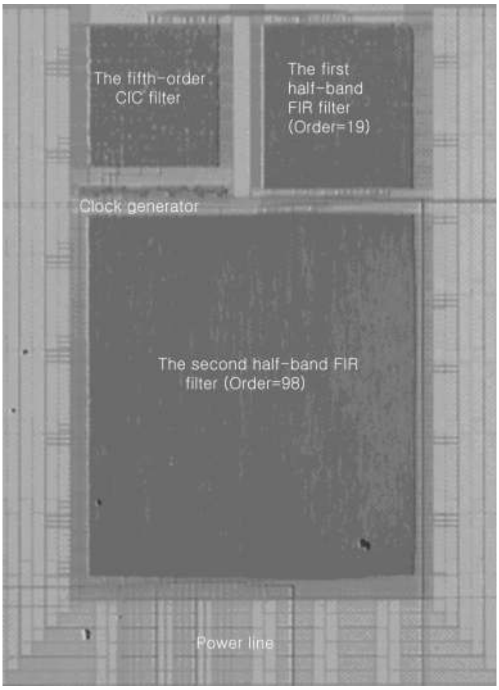


Figure 9. Die photograph of the decimation filter.

Table 4. Performance of the digital decimation filter.

Input data rate	2.8224 MHz
Output data rate	44.1 MHz
Passband	0–20 kHz
Stopband	> 24.1 kHz
Minimum stopband attenuation	80 dB
Phase distortion	0° (linear phase)
Area	1.36 mm^2
Technology	0.25-μm CMOS

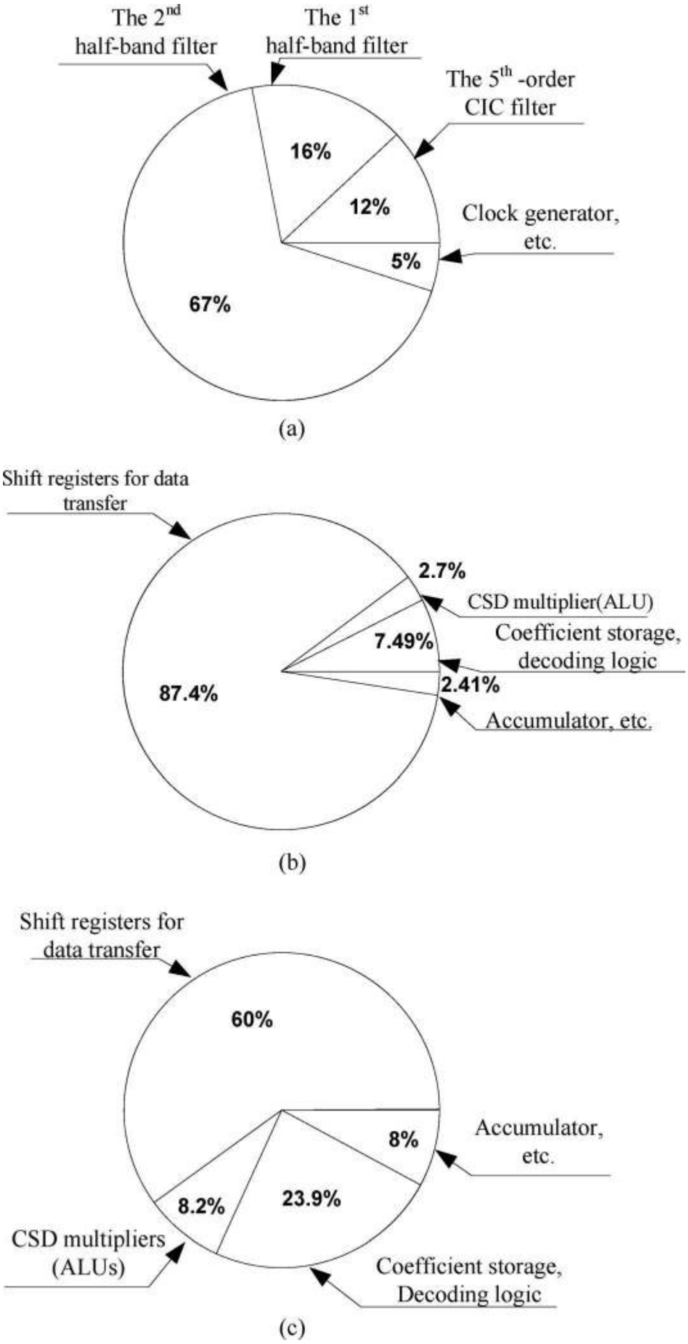


Figure 10. The area analysis of (a) the digital decimation filter, (b) the first half-band FIR filter, and (c) the second half-band FIR filter.

than 80 dB. The magnified passband band characteristic shows about −6 dB passband droop at the edge of the passband. As mentioned earlier, a simple extra droop-compensation filter might be required. The maximum operating frequency of the filter is simulated after the logic synthesis. The nominal clock frequency is 2.8224 MHz, but the simulated maximum frequency is 50 MHz when we increased the clock frequency in 1-MHz steps with the output data rate of 781.25 kHz. This shows that the design procedures in this paper can be used in even higher speed applications.

3. Experimental results

The digital decimation filter is fabricated in a 0.25-μm CMOS process and has an active area of 1.36 mm ^{2} (940 × 1450 μm). Figure 9 shows a die photograph of the decimation filter. The clock generator is located between the fifth-order CIC filter and the second half-band FIR filter to provide the shortest distances for the three kinds of clocks. Table 4 shows the performance of the implemented digital decimation filter. This digital decimation filter is implemented to remove the out-of-band noise caused by delta-sigma modulation and to reduce the input sampling rate to the final Nyquist rate. The digital decimation filter has a passband of 20 kHz and a minimum stopband attenuation of 80 dB for digital audio applications. The final resolution of the digital decimation filter is 23 bits.

Figure 10 illustrates the area analysis of the fully synthesised decimation filter. The second half-band FIR filter, the first half-band FIR filter, and the fifth-order CIC filter occupy 67%, 16%, and 12% of the active area, respectively. Within the second half-band FIR filter, which occupies the largest area overall, blocks for coefficient storage and decoding logic occupy 23.9% while the ALU occupies 8.2 %. The extra area contains the shift registers for data transfer. Results show that the second half-band FIR filter occupies the largest area because it has the largest number of multiplication and addition operations.

Blocks for the storage of coefficients and decoding logic, having an equivalent function to ROM, occupy 16% of the digital decimation filter. The area occupied by shift registers, which are used for data transfer in this design instead of RAM, are the largest overhead in a fully synthesised chip without ROM and RAM.

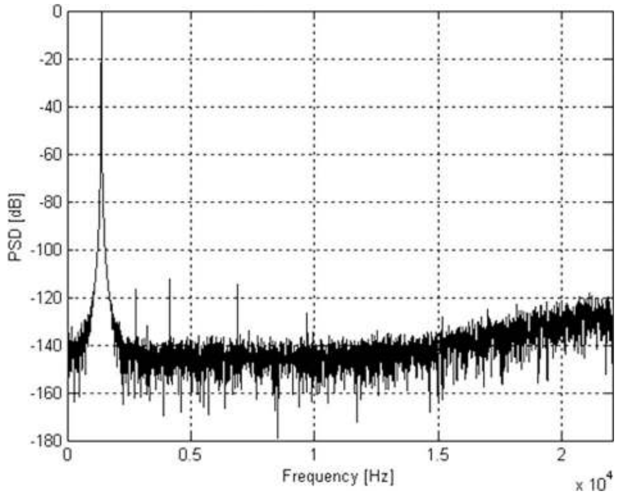


Figure 11. The 13 kHz, −0.5 dB input (8192-point FFT).

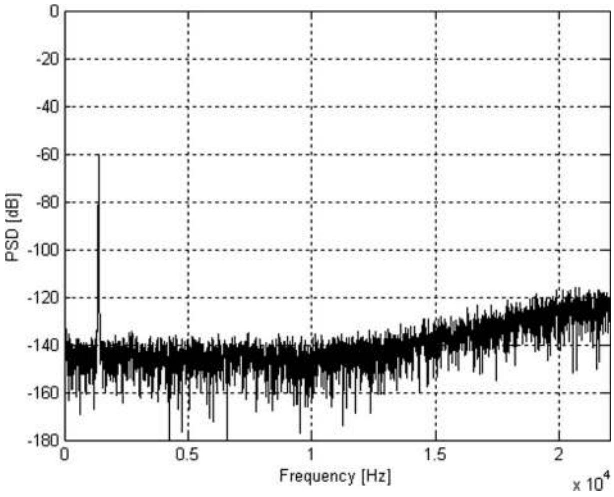


Figure 12. The 13 kHz, −60 dB input (8192-point FFT).

Figures 11 and 12 show the measured output spectrum in the baseband, with 8192 output samples of the digital decimation filter captured by the Logic Analyzer, where the input amplitudes are −0.5 dB and −60 dB, respectively. The measured output spectra show that the signal component in the baseband is passed without loss. Most of the quantization noise caused by delta-sigma modulation is attenuated and the final output frequency band of the digital decimation filter is limited to the audio signal band (22.05 kHz).

4. Conclusion

This paper presents the design and implementation of a digital decimation filter with three stages (CIC-FIR-FIR) for high-resolution delta-sigma applications. This filter provides a time-to-market advantage. Multiplications of CSD-encoded coefficients are implemented by using shifters and adders. The complete functions, including the storage of coefficients and computation, are implemented by using Verilog-HDL synthesis instead of RAM and ROM. Experimental results show that most of the out-of-band quantization noise caused by delta-sigma modulation is significantly attenuated without aliasing. The presented decimation filter is adequate for use in digital audio, portable electronics, and telecommunication applications.

Acknowledgements

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (No. 20100000203).

References

Schreier, R., and Temes, G.C. (2005), *Understanding Delta-Sigma Data Converters*, NJ: IEEE Press.

Roh, J., Byun, S., Choi, Y., Roh, H., Kim, Y., and Kwon, J. (2008), 'A 0.9-V 60 μW 1-bit 4th-order delta-sigma modulator with 83-dB dynamic range', *IEEE Journal of Solid-State Circuits*, 43, 361–370.

Roh, H., Kim, H., Choi, Y., Roh, J., Kim, Y., and Kwon, J. (2009), 'A 0.6-V delta-sigma modulator with subthreshold-leakage suppression switches', *IEEE Transactions on Circuits and Systems II*, 56, 825–829.

Brandt, B.P., and Wooley, B.A. (1994), 'A low-power, area-efficient digital filter for decimation and interpolation', *IEEE Journal of Solid-State Circuits*, 29, 679–687.

Meyer-Baese, U. (2004), *Digital Signal Processing with Field Programmable Gate Arrays*, 2nd ed., London: Springer-Verlag.

Yang, Y., and Fathy, A.E. (2009), 'Development and implementation of a real-time see-through-wall radar system based on FPGA', *IEEE Transactions on Geoscience and Remote Sensing*, 47, 1270–1280.

Dolecek, C.J., and Mitra, SK. (2005), 'A new sharpened cascaded comb-cosine decimation filter', in *13rd European signal processing conference*, Antalya, Turkey, September 4–8 2005.

Dolecek, C.J., and Mitra, SK. (2008), 'On design of CIC decimation filter with improved response', in *3rd International Symposium on Communications, Control and Signal Processing*, pp. 1072–1076.

Laddomada, M. (2008), 'Design of multistage decimation filters using cyclotomic polynomials: Optimization and design issues', *IEEE Transaction on Circuits and Systems I*, 55, 1977–1987.

Gao, Y., and Tenhunen, H. (1999), 'Low-power implementation of a fifth-order comb decimation filter for multi-standard transceiver applications', in *Personal Computing and Communication*, Lund, Sweden, 29–30 November 1999.

Goodman, D.J., and Carey, M.J. (1977), 'Nine digital filters for decimation and interpolation', *IEEE Transactions on Acoustics, Speech, and Signal Processing,* ASSP-25, 121–126.

Hogenauer, E.B. (1981), 'An economical class of digital filters for decimation and interpolation', *IEEE Transactions on Acoustics, Speech, and Signal Processing,* ASSP-29, 155–162.

Harris, F.J. (2004), *Multirate Signal Processing for Communications Systems*, NJ: Prentice Hall.

Hewlitt, R.M., and Swartzlander, E.S. (2000), 'Canonical signed digit representation for digital filters', in *IEEE Workshop on Signal Processing Systems*, pp. 416–426.

Candy, J. (1986), 'Decimation for sigma delta modulation', *IEEE Transactions on Communications,* COM-34, 72–76.

Proakis, J.G., and Manolakis, D.G. (1996), *Digital Signal Processing Principles, Algorithms, and Applications*, 3rd ed., NJ: Prentice Hall.