Compressed fixed-point data formats with non-standard compression factors

Manuel Richey

Engineering Services Department,
CertTech (certtech.com),
Lenexa, KS, 66215 USA
Email: mrichey@certtech.com

Hossein Saiedian*

Electrical Engineering and Computer Science,
Information & Telecommunication Technology Center,
University of Kansas (eecs.ku.edu),
Lawrence, KS, 66049 USA
Email: saiedian@eecs.ku.edu
*Corresponding author

Abstract: Sign bit compression in fixed-point numbering systems can improve the dynamic range and round-off noise for signal processing algorithms. This paper analyses non-standard compression factors (CF) for compressed fixed-point data formats, where sign bit compression is performed on each individual fixed-point number. Although these compression techniques are applicable to other fixed-point formats, the compressed two's complement data format is selected for illustration. A brief background on compressed two's complement is provided. Obvious compression factors are powers of two due to binary formatting, but compression factors other than standard powers of two are presented. Compression factors of 3 and 5 are analysed in greater detail. Motivation for and advantages of non-power-of-two compression factors are identified.

Keywords: compression factors; digital signal processing; fixed-point data formats; multiple shift size mapping; sign bit compression; signal processing; variable shift field length.

Reference to this paper should be made as follows: Richey, M. and Saiedian, H. (2017) 'Compressed fixed-point data formats with non-standard compression factors', *Int. J. Signal and Imaging Systems Engineering*, Vol. 10, No. 6, pp.301–305.

Biographical notes: Richey is a Senior Staff Engineer at CertTech (certtech.com) in Kansas. He has a Master's degree from the University of Kansas. He has a number publications (primarily in IEEE journals) and several US patents.

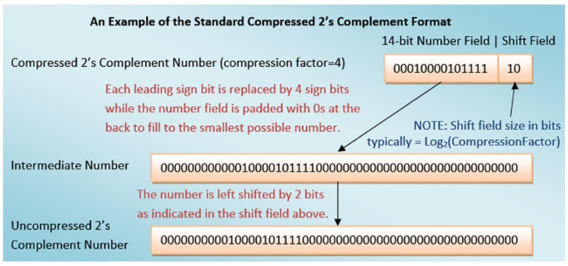
Professor Saiedian (Ph.D., 1989) is a Professor at the Department of Electrical Engineering and Computer Science. His research has been supported by the NSF and other federal units. Saiedian has over 175 publications.

1 Background on the compressed two's complement data format

Techniques for compressing the fixed-point data format are broadly applicable to most fixed-point formats, but we will illustrate compression for the two's complement format due to the traditional advantages it has over the other fixed-point data formats (Parhami, 1999; Goldberg, 1991). The compressed two's complement data format has been described in some detail by the authors in previous publications (Richey and Saiedian, 2009; Richey and Saiedian, 2011). Here, we will provide a brief description of the format to introduce the new material provided in this paper. The compressed two's complement data format compresses the sign bits of a standard two's complement data format by a compression factor (CF) that is assumed in the implementation, but not encoded into the data.

A shift field is added to the format to allow coverage of the entire numeric range without holes. As an example, a compressed two's complement number with a compression factor of 4 would expand each leading sign bit from one binary digit into four binary digits. The extra space is used to allow additional bits of precision for each number. This format is illustrated in Figure 1. The shift field at the end of the number indicates how many bits to shift the number to the left after decompression of the sign bit. So, every sign bit in the numeric field is expanded to four sign bits and then the resulting number is shifted to the left by the number of bits indicated in the shift field (0, 1, 2 or 3 bits).

Figure 1 Decompression of a compressed two's complement number (compression factor = 4) (see online version for colours)



Compressed two's complement numbers not only provide large dynamic range, like floating-point formats, but also provide uneven numeric precision, like fixed-point formats. This is an advantage for some problem domains such as digital signal processing (Richey and Saiedian, 2009). This is the case because in digital signal processing systems that utilise fixed-point numbering, automatic gain control is typically used to move the signal to the top of the numeric range for the given data format.

2 Non-standard compression factors

As can be seen from Figure 1, a typical compression factor would require a fixed number of bits for the shift field. For example, a compression factor of 4 would require a shift field of two bits ($\log_2$(compression factor) = $\log_2(4) = 2$). These standard compression factors would all be powers of two (i.e. 1, 2, 4, 8 …). Compression factors that are not a power of two may at first appear to be at a disadvantage because the shift bits would not be optimally organised. Thus, a compression factor of 9 would still require 4 bits for the shift field just like a compression factor of 16, but would have a significant disadvantage in dynamic range when compared to a compression factor of 16.

However, if organised properly, the unused bits from a non-power-of-two compression factor can still be used for additional precision. This is accomplished by allowing the shift field to vary in width. Compression factors that employ non-uniform mechanisms such as varying shift field widths are referred to in this paper as non-standard compression factors.

3 Compression factor of 3

The simplest example of a non-standard compression factor is illustrated by a compressed two's complement number with a compression factor of 3 (Figure 2). This figure shows that a compression factor of 3 provides an additional bit of precision for a third of the numbers when compared to a compression factor of 4. Furthermore, by placing the greater precision with the largest number of left shifts, the additional precision is placed in the most important numeric position. With a compression factor of 3, the dynamic range is extended close to three times the uncompressed range, and noise is approximately on par with compression factor of 2. This technique is called *variable shift field length*.

Figure 2 A compressed two's complement format with a compression factor of 3 (see online version for colours)

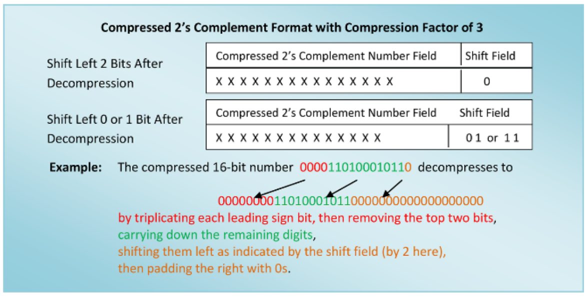
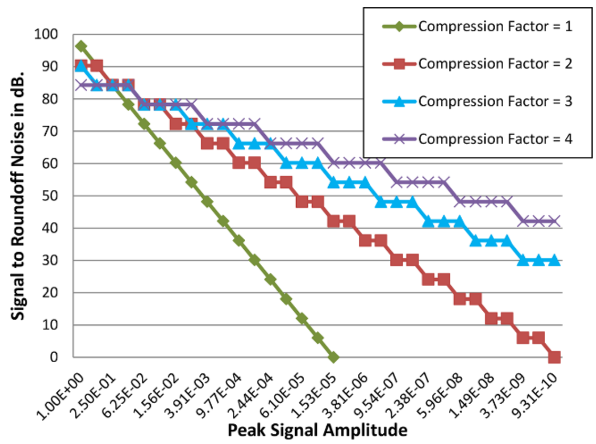


Figure 3 plots the precision for compression factors of 1 (regular 2's complement), 2, 3 and 4 for 16-bit numbers. The chart shows that for a compression factor of 3, only one binary range is reduced in precision compared to a compression factor of 2, while the majority of lower level binary ranges are improved. Compared to a compression factor of 4, a compression factor of 3 achieves greater precision for the largest numbers. For digital signal processing algorithms, this can result in improved noise performance. These differences may provide some incentive for using a compression factor of 3 as opposed to one of either two or four.

Figure 3 Peak signal vs. peak round-off noise for several small compression factors in fractional compressed two's complement format (see online version for colours)

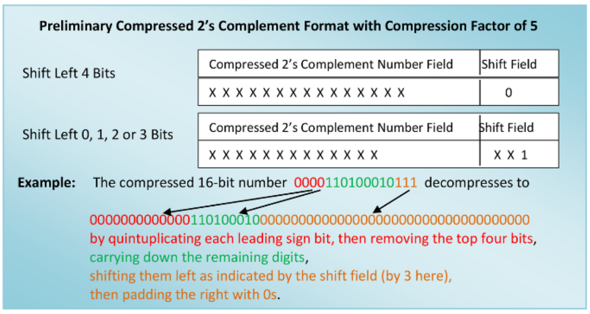


Note: CF of 1 is standard 2s complement

4 Compression factor of 5

The same technique that was used to create a compression factor of 3 can be used for a compression factor of 5 (Figure 4). However, since we have more than one bit of difference between the shift field sizes, the concentration of precision will not be even (Figure 6a). Concentration of precision represents the main advantage compressed two's complement has over floating point. So, we can preserve this with the use of another technique for compression factor of 5.

Figure 4 A compressed two's complement format with a compression factor of 5 (see online version for colours)



The technique we will use to focus the precision for a compression factor of 5 is to make the upper format shift field indicate a variable amount of left shift. For the largest precision numbers (i.e. those with only one sign bit and a shift field of 0), the upper format from Figure 4 will shift 5 bits to the left. However, for all other numbers with a shift field of 0, it will shift 9 bits to the left as shown in Figure 5. All numbers with a shift field greater than 0 shift normally (i.e. 0, 1, 2 or 3 bits). This technique will focus the precision for compression factor of 5 in a regular manner as shown in Figure 6b. This technique is called *multiple shift size mapping*.

Figure 5 An adjusted compressed two's complement format with a compression factor of 5 (see online version for colours)

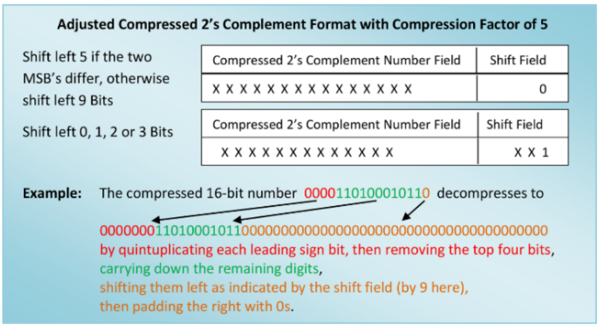
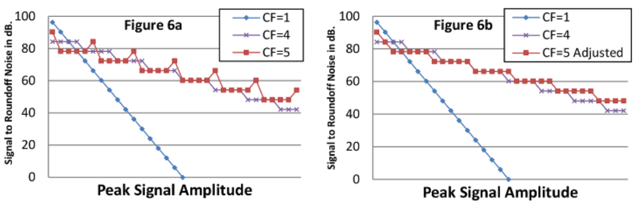


Figure 6 A comparison of compression factors 1, 4 and 5 in fractional compressed two's complement format (6a is pre-adjustment, whereas 6b is post-adjustment) (see online version for colours)



5 Motivation: benefits of non-standard compression factors

Signal processing systems are under constant pressure to improve performance and reduce cost. Compression techniques are often applied to signals (Kanhe and Hamde, 2016; Kumari and Rajalakshmi, 2016) rather than data formats to accomplish this goal. Nevertheless, significant effort is typically applied to select the most advantageous data format to implement a system or algorithm (Darulova and Kuncak, 2017). Numerous non-traditional data formatting mechanisms have been considered for use with signal processing applications to overcome the simultaneous problems of small dynamic range and round-off noise (Ray, 2010). Several have actually demonstrated that non-standard mechanisms for data formatting can improve noise performance over traditional fixed and floating-point formats (Azmi and Lombardi, 1989; Mishra and Jena, 2011). A few even focus on sign bit compression (Koyama et al., 2012) similar to the formats shown in this paper. It has also been shown that compressed two's complement data formats outperform both traditional fixed-point and floating-point data formats in terms of noise performance for signal processing applications (Richey and Saiedian, 2011).

In this paper, we have demonstrated that non-standard compression factors (i.e. those that are not powers of two) are also viable options. We now need to show that these non-standard compression factors can provide a computational advantage.

Using our two techniques (*variable shift field length* and *multiple shift size mapping*), data formats with a compression factor of 5 can outperform the standard $\log_2$ compression factor of 4. A compression factor of 5 can simultaneously provide greater precision for the largest numbers, and greater precision for smaller numbers than a compression factor of 4 provides. This is observed in Figure 6b which presents a comparison between these two compression factors.

An impressive way to look at the advantages of non-standard compression factors is by comparison with traditional two's complement formats. A standard 16-bit two's complement integer can range from –32768 to +32767 with no fractional component. If one adds a single bit of word width to the format and then compresses the sign bits with a compression factor of either 3 or 5, they will still cover every single integer value from –32768 to +32767. However, one also obtains 42 additional bits of dynamic range (with compression factor of 5) on the fractional side of the decimal point. This will noticeably improve round-off noise performance and allow about five times as much dynamic range in decibels without giving up anything more than a single bit of word width. Of course, IEEE 754 floating point achieves over twice this improvement in dynamic range in dB, but at the cost of doubling the word width to 32 bits (Richey and Saiedian, 2011).

Obviously, one does not have to grant that extra bit of word width to obtain a massive benefit. Without expanding the size of the word, one can still obtain vastly greater dynamic range and improved noise performance with compressed two's complement than one would obtain with the same sized word in regular two's complement. In the case of a 16-bit compressed two's complement number format with a compression factor of 5 and appropriate placement of the decimal point, every integer between 16383 and –16384 is represented in the data format, which results in a 6 dB loss of dynamic range above the decimal point. However, 43 bits of dynamic range is provided below the decimal point for this format. Thus, the total dynamic range of the format is significantly increased. If this increase in dynamic range is accompanied with the use of automatic gain control techniques that are typically used in traditional fixed-point signal processing systems, then most algorithms will experience an improvement in both dynamic range and round-off noise.

6 Experimental confirmation

To verify the asserted performance improvements of irregular compression factors, a data format with a compression factor of 3 was implemented and tested against previously coded formats with compression factors of 1 and 2 (Richey and Saiedian, 2011). Note that a compression factor of 1 results in standard two's complement. In the testbed, a two-toned signal is passed through a Hanning window and a 1024 point Fast Fourier transform. This is done for all three formats with the tone frequencies varying for each format to increase waveform visibility, and each second tone being attenuated by 40 dB in relation to the first. The compressed data formats were uncompressed for computation, but then recompresses for storage into memory. The uncompressed format (CF=1) was merely rounded prior to storage into memory. Since no additional noise was added to the system, recompression and rounding are the principal sources of noise for this simulation.

The advantage of format compression is illustrated in Figures 7 and 8. In Figure 7, the primary tone is near peak amplitude for these formats. As can be seen, the noise floor drops significantly between the uncompressed format and the two compressed formats. Figure 7 shows that although a compression factor of 3 provides a significant improvement in dynamic range (about 84 dB.) over a compression factor of 2, no degradation in noise performance is observed for maximum amplitude signals.

Figure 7 Compression factor comparison in the frequency domain on maximum amplitude 16-bit two-toned signals (processing includes a Hanning Window and 1024 Point FFT for compression factors of 1, 2 and 3) (see online version for colours)

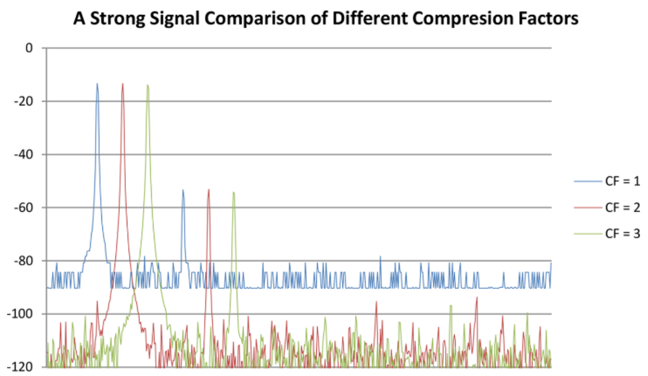
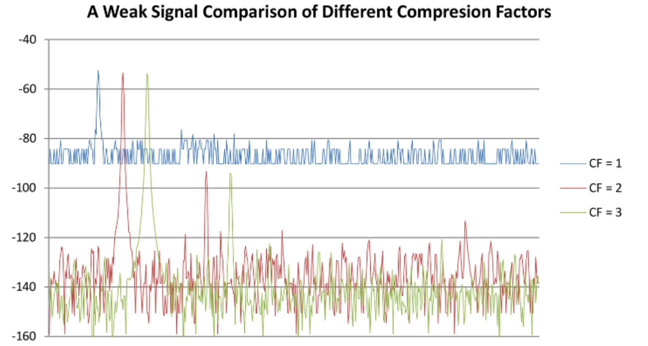


Figure 8 Compression factor comparison in the frequency domain on minimal amplitude 16-bit two-toned signals (processing includes a Hanning Window and 1024 Point FFT for compression factors of 1, 2 and 3) (see online version for colours)



Of course, a maximum amplitude signal represents the worst case comparison for a compression factor of 3. As the primary carrier power drops, the data format with a compression factor of 3 improves substantially when compared to both a compression factor of 2 and also to standard fixed point (CF = 1). In Figure 8, we see the performance of these same formats with the input signal attenuated by 40 dB. In this chart, the second tone does not really show up with the fixed-point format due to dynamic range limitation. The CF = 3 format in Figure 8 has an average improvement in the noise floor of greater than 5dB over the CF = 2 format. Of course, this improvement increases as the signals become even smaller. Although we did not simulate the compression factor of 5, these same trends would be evident with that format when compared with compression factors of 1, 2 and 3. As Figures 7 and 8 illustrate, compressed two's complement data formats provide the important advantage of data scaling as found in floating-point formats, but overcome the severe disadvantage standard floating point incurs by allocating a fixed number of bits to an exponent field. Irregular compression factors such as 3 and 5 enhance the advantages of compressed two's complement.

Of course, rounding is very important in small data formats, and traditional rounding techniques are well understood for fixed-point systems (Kuck et al., 1977; Goldberg, 1991; Menard et al., 2006; Kim et al., 1998). However, an analysis of rounding techniques for sign bit compressed formats has not been explored in depth and sophisticated rounding may actually improve the performance of these formats over that shown in Figures 7 and 8.

7 Summary

Non-standard compression factors provide interesting and possibly optimal formatting mechanisms for digital signal processing applications. However, non-standard compression factors require additional techniques over normal power-of-two compression factors to achieve appropriate mapping of precision. The two techniques discussed in this paper address formatting for compression factors of 3 and 5. The two techniques presented here are as follows:

1 Variable shift field length.

2 Multiple shift size mapping.

Similar techniques are required to adjust other non-power-of-two compression factors for concentration of precision.

We have pointed out advantages of using non-standard compression factors, and shown that a compression factor of 3 has some advantages over its neighbouring compression factors of 2 and 4. We have also shown that a compression factor of 5 has precision advantages for both larger and smaller numbers over a compression factor of 4. These advantages may allow non-power-of-two compression factors to become optimal solutions for many applications.

References

Azmi, A. and Lombardi, F. (1989) 'On a tapered floating point system', *Proceedings of the 9th Computer Arithmetic Symposium*, IEEE Computer Society Press, Washington, D.C., pp.2–9.

Darulova, E. and Kuncak, V. (2017) 'Towards a compiler for reals', *ACM Transactions on Programming Languages and Systems*, Vol. 39, No. 2. pp.1–28.

Goldberg, D. (1991) 'What every computer scientist should know about floating-point arithmetic', *ACM Computing Surveys*, Vol. 23, No. 1, pp.5–48.

Kanhe, R. and Hamde, S. (2016) 'ECG signal compression using 2-D DWT Hermite coefficients', *International Conference on Signal and Information Processing (IConSIP)*, Vishnupuri, India.

Kim, S., Kum, K. and Sung, W. (1998) 'Fixed-point optimization utility for C and C++ based digital signal processing programs', *IEEE Transactions on Circuits and Systems*, Vol. 45, No. 11, pp.1455–1464.

Koyama, J., Yamori, A. Kazui, K., Shimada, S. and Nakagawa, A. (2012) 'Coefficient sign bit compression in video coding', *Picture Coding Symposium*, Krakow, Poland.

Kuck, D., Parker Jr., D., and Sameh, A. (1977) 'Analysis of rounding methods in floating-point arithmetic', *IEEE Transactions on Computers*, Vol. 26, No. 7, pp.643–650.

Kumari, R. and Rajalakshmi, E. (2016) 'Development of wavelet based signal compression algorithm using adaptive coder in FPGA', *Online International Conference on Green Engineering and Technologies (IC-GET)*, Coimbatore, India.

Menard, D., Chillet, D., and Sentieys, O. (2006) 'Floating-to-fixed-point conversion for digital signal processors', *EURASIP Journal on Applied Signal Processing*, Vol. 2006, No. 1, pp.1–19.

Mishra, A. and Jena, S. (2011) *Performance Evaluation of Orthogonal Frequency Division Multiplexing using 16-Bit Irregular Data Formats*, Thesis, Department of Electrical & Communication Engineering, National Institute of Technology, India.

Parhami, B. (1999) *Computer Arithmetic: Algorithms and Hardware Designs*, Oxford University Press, New York.

Ray, G. (2010) 'Between fixed and floating point', *Chip Design: Tools, Technologies & Methodologies*, pp.17–22.

Richey, M. and Saiedian, H. (2009) 'A new class of floating-point data formats with application to 16-bit digital-signal processing systems', *IEEE Communications*, Vol. 47, No. 7, pp.94–101.

Richey, M. and Saiedian, H. (2011) 'Compressed two's complement data formats provide greater dynamic range and improved noise performance', *IEEE Signal Processing*, Vol. 28, No. 6, pp.154–158.