

Effect of Rounding and Saturation in Fixed-Point DSP Implementation of IFFT and FFT for OFDM Applications

Jean Armstrong¹
Monash University
Dept. of Elect. & Comp. Sys. Eng.
Victoria, 3800, Australia.
+61 3 9905 5355
Jean.Armstrong@eng.monash.edu.au

Himal A. Suraweera¹
Monash University
Dept. of Elect. & Comp. Sys. Eng.
Victoria, 3800, Australia.
+61 3 9905 5355
Himal.Suraweera@eng.monash.edu.au

Simon Brewer
Analog Devices Australia P/L
Unit 3, 97 Lewis Road
Wantirna, Victoria, 3152
Australia. +61 3 9881 9902
simon.brewer@analog.com

Robert Slaviero
Analog Devices Australia P/L
Unit 3, 97 Lewis Road
Wantirna, Victoria, 3152
Australia. +61 3 9881 9900
robert.slaviero@analog.com

Abstract -- This paper describes a new technique for designing FFTs for Orthogonal Frequency Division Multiplex (OFDM) transmitters and receivers. Using the new technique, signals within the FFT structure are scaled so that there is an optimum trade-off between saturation and rounding. Conventional FFT designs avoid saturation, either by using fixed scaling factors which ensure that signal levels are always below the saturation level or by using block floating point. For fixed-point arithmetic of a given precision, the new technique results in better signal to noise ratio (SNR). The SNR improvement increases with the size of the FFT. The implementation complexity is significantly less than that of block floating point and equal, or slightly greater than that of conventional fixed scaling, depending on the radix of the FFT. Matlab simulation results are presented which compare the performance of the new technique with conventional designs for a range of FFT sizes. An FFT using the new design has been implemented and tested using the Analog Devices ADSP-BF533 Processor demonstrating the performance of the new technique in a real-time environment.

I. INTRODUCTION

Orthogonal frequency division multiplexing (OFDM) is the modulation technique used in many new and emerging broadband communication systems including wireless local area networks (LANs), high definition television (HDTV) and 4G systems [1]. The key component in an OFDM transmitter is an inverse fast Fourier transform (IFFT) and in the receiver, an FFT. The increasing computational power and performance capabilities of DSPs make them ideal for the practical implementation of OFDM functions. Consumer products are usually sensitive to cost and power consumption and for this reason, a fixed-point DSP approach is preferred. However, fixed-point systems have limited dynamic range, causing the related problems of round-off noise and arithmetic overflow.

The traditional approach to FFT design is to scale the signal so that overflow is avoided [2]. For OFDM systems using larger FFTs and fixed-point implementation, a large word length is required if rounding errors are not going to significantly degrade system performance. Scaling is usually distributed throughout the FFT structure as this reduces the overall effect of rounding errors [2]. To completely eliminate the possibility of overflow in a radix-2 implementation, numbers must be scaled by a factor of one half after each butterfly stage. Another way to avoid overflow is to use Block

Floating Point (BFP) scaling [3]. This adapts the scaling at each stage of the FFT according to the data, so that overflow does not occur for the given input data. However a single large signal sample can result in scaling which causes large round off errors in all of the other signal values. To avoid this problem some researchers have proposed a compromise method called convergent BFP, where differing scaling factors are used for different sections of data [4].

In this paper we describe a new approach which is better suited to FFT design as applied to multicarrier modulation systems such as OFDM. The signals are scaled so that overflow, rather than being completely avoided, occurs with low probability throughout the IFFT and FFT structures. The size of the error that results from an overflow depends on how overflow is handled in the DSP. To minimize the degradation, overflow should result in saturation of the value at the maximum positive or negative value option. This is equivalent to clipping the signal. Using the new technique, signals within the FFT structure are scaled to balance the effect of clipping and round-off. Clipping may result in comparatively large errors in a few signal values but because OFDM systems typically include error coding/correction, system performance depends on the total error or, in other words the total noise power, across all of the FFT outputs rather than on any individual value.

In order to understand and analyze the new technique, background information is required on a number of different topics. These are:

- OFDM, and in particular the effects of noise in OFDM;
- FFT structures and fixed-point implementation and the effect of limited arithmetic precision;
- the statistics of signals within the IFFT/FFT in OFDM;
- and the quantization of Gaussian signals.

This background is given in the following sections, then simulation results are presented for the new technique and compared with the results for conventional techniques. The practical implementation of the new technique on a fixed-point DSP is also presented.

II. DESCRIPTION OF OFDM SYSTEM

Fig. 1 shows a simplified block diagram of an OFDM transmitter and receiver. The data to be transmitted is first serial-to-parallel converted and then mapped onto complex numbers representing values from the signal constellation

¹ Formerly with La Trobe University when this research was performed.

which is being used. Each input controls the signal at one frequency. The IFFT performs, in one operation, the modulation of each subcarrier and the multiplexing of these subcarriers. The signal is then parallel-to-serial converted, converted to analog, and modulated onto a high frequency carrier. The details of how this is achieved and the order of the blocks may vary in practice.

At the receiver the signal is downconverted, converted to digital, and serial-to-parallel converted, before being input to the FFT. The FFT performs demodulation and demultiplexing of each subcarrier. The distortion in the channel has the effect of changing the phase and amplitude of each subcarrier. This is corrected by the single tap equalizer. The data is then input to the error decoder. The presence of the error decoder means that the overall error rate does not depend on the noise in any one of the FFT outputs, but on the statistics of the noise across all FFT outputs. This is closely related to the ‘noise bucket’ effect which has been observed for impulse noise in OFDM [5].

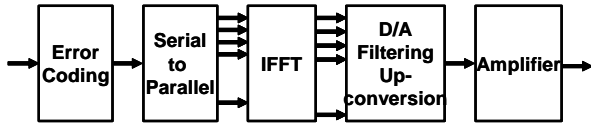


Fig. 1 (a) Block diagram of a typical OFDM transmitter.

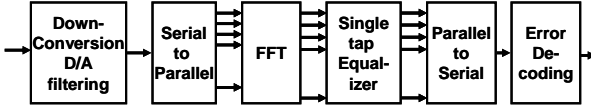


Fig.1 (b) Block diagram of a typical OFDM receiver.

III. FFT STRUCTURES AND FIXED-POINT IMPLEMENTATION

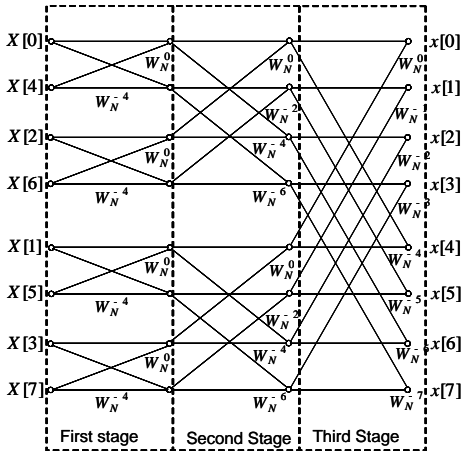


Fig.2. Butterfly structure for an 8 point radix-2 transform.

The FFT algorithm is based on a butterfly structure. Fig. 2 shows the structure of an 8-point radix-2 transform. It has three stages. Each butterfly involves multiplication and addition. When fixed-point arithmetic is used, rounding and possibly overflow occur. The precise points within the butterfly at which these may occur depends on details of the implementation. Scaling factors may also be used between stages. In the Matlab simulations, the structure in Fig. 3 was used to model each stage. Quantization was modeled only at

the output of each stage. Infinite precision was assumed within each stage.

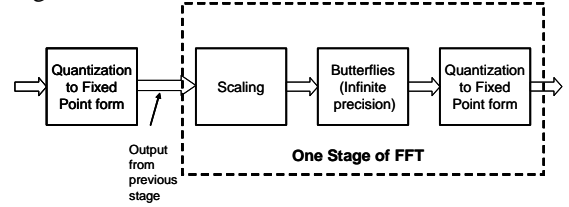


Fig. 3. Model of each FFT stage used in the Matlab simulations.

IV. STATISTICS OF THE SIGNALS WITHIN THE IFFT AND FFT

The signals within the IFFT and FFT in an OFDM system are complex random variables. The effect of rounding and clipping depends on the probability distribution of the signals throughout the FFT and IFFT structures. Fig. 4 shows the complementary cumulative distribution (CCD) for the real components at various stages in the transmitter IFFT structure for a 64-point radix-2 transform and 4 QAM inputs. No scaling is used, so the mean square (MS) values of the signals grow at each stage as signals from the previous stages are summed. For the first three stages, the outputs clearly do not have a Gaussian distribution, however by the fourth stage, the tail of the distribution is quite close to Gaussian and by the fifth stage the difference from a Gaussian distribution is negligible.

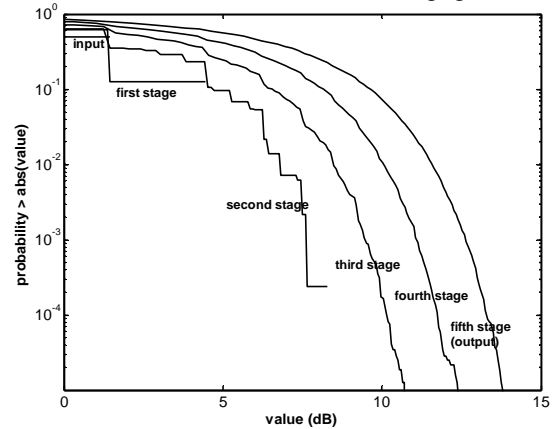


Fig. 4. Complementary cumulative distribution of real components within the IFFT structure for 4 QAM, 64 point FFT, radix-2 transform.

Another important point to note in Fig. 4 is that when the input to any stage is Gaussian, the output is Gaussian with an increase in power of 3dB. This means that the average power has increased by a factor of two and the amplitude by $\sqrt{2}$. So that although the *maximum* signal increases by a factor of two, the *average* amplitude increases by $\sqrt{2}$. This means that the use of scaling factors of 0.5 results in a decrease in average power by 3dB at each stage.

V. QUANTIZATION OF GAUSSIAN RANDOM VARIABLES

Quantization of Gaussian random variables has been extensively studied in the past both in general [6] and in the context of design of analog-to-digital converters (ADCs) for OFDM [7]. These results can be directly applied to the problem of saturation and rounding in fixed-point arithmetic.

To describe the properties of a uniform quantizer, three related parameters must be defined. These are d , the distance between quantization levels, k , the number of quantization levels, and h_r , the *headroom* which is the maximum quantization level. Throughout this paper h_r is normalized to the root mean square (RMS) value of the signal. When quantization is symmetrical about zero, the three parameters are related by

$$h_r = \frac{(k-1)d}{2} \quad (1)$$

Quantization and clipping noise have very different statistical properties. In an OFDM system, clipping will be a relatively rare event but most samples will be subject to quantization noise. Usually the quantization noise is approximately uniformly distributed between $-d/2$ and $+d/2$. For a given h_r , if k is doubled, the quantization noise will be reduced by 6 dB. Similarly if k is kept constant and h_r is increased by 6 dB, the quantization noise will increase by 6 dB. Clipping noise has more complicated characteristics. It is impulsive in form. The impulses of clipping noise occur infrequently but may have amplitudes much larger than quantization noise. As h_r increases, the probability of clipping decreases.

Fig. 5 shows the complementary cumulative distribution for the error in quantizing a Gaussian variable when $h_r = 12$ dB and $k = 2^{16}$, i.e. 16 bits are used to represent each value. The graph has two distinct regions. The region at the left is the result of the quantization noise. The region to the right is the result of clipping noise. The noise to signal level at the transition between regions depends on d . The level of the 'plateau' indicates the probability of clipping, so increasing h_r makes the plateau occur at a lower probability. However, because of the infinite tail of the Gaussian distribution, there is always a non-zero probability of clipping and the graph will have the same basic form irrespective of the value of the parameters.

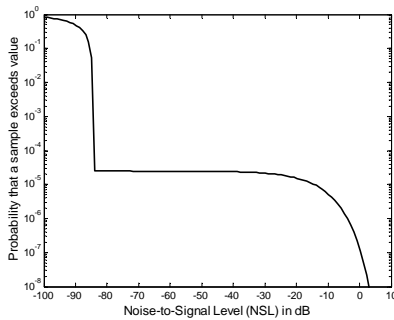


Fig. 5. Complementary cumulative distribution of noise to signal level for quantization of Gaussian random variable with $h_r = 12$ dB and $k = 2^{16}$.

Fig. 6 shows the mean square error (MSE) versus headroom for quantization of Gaussian random variables, with varying number of bit resolution. It shows that for each number of bits there is a headroom which gives the minimum mean square error (MMSE). However the MMSE headroom does not necessarily give best system performance in OFDM. The effect of low probability, high noise and clipping events must also be considered.

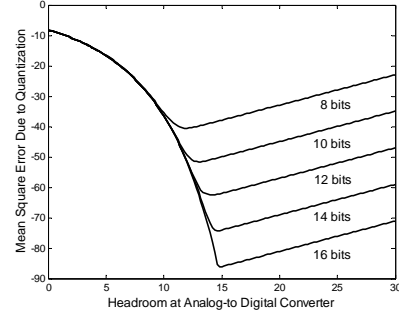


Fig. 6. Error in quantizing Gaussian random variables versus headroom.

VI. FFT/IFFT DESIGN WITH CONTROLLED CLIPPING

The information in the previous sections will now be brought together to show how an FFT with controlled clipping can be designed for a particular OFDM application.

To summarize the key results of the previous sections:

- The performance for an OFDM symbol depends on the statistics of the total noise added to the symbol;
- The signals at many points within the IFFT and FFT have a Gaussian distribution;
- When a Gaussian signal is quantized there is a trade-off between clipping errors and rounding errors which depends on the headroom of the quantizer;
- At each stage within an FFT the signal power doubles, so the average amplitude increases by $\sqrt{2}$.

The optimum design and selection of scaling adjusts the signal power throughout the FFT/IFFT so that the headroom at each point gives the optimum compromise between clipping and rounding. The optimum value depends on many aspects of the overall system design, including the number of subcarriers, the power and statistics of other sources of noise, and the number of bits used in fixed-point representation. For Gaussian signals the optimum headroom is generally slightly above the headroom for MMSE. To maintain the optimum headroom throughout the transform, the signal power should be kept constant or approximately constant. For a radix-2 based structure, the optimum scaling factor is therefore $1/\sqrt{2}$ at each stage. A slightly suboptimum structure, which is more readily implemented on a DSP, is to scale by 1 and $1/2$ (simple shift left) at alternate stages.

VII. SIMULATION RESULTS

Matlab simulations were used to examine the performance of OFDM transmitters and receivers incorporating IFFTs and FFTs designed using the new technique, and were compared with the performance using conventional fixed scaling and block floating point implementations. Fig. 7 shows the model used in the receiver simulations. To understand the effect of FFT design on overall system performance, we must take into account how the signals are represented in fixed-point form at the input to the FFT. There is no point in designing an FFT to avoid clipping, when inevitably the signal at the input has already been limited by the finite range of the analog to digital

conversion (ADC). In Fig. 7 this is indicated by the ‘ADC’ block. However, in practice, this conversion from high frequency analog to baseband digital will involve a number of functions. The amount of clipping which will occur in the FFT depends directly on the headroom of the ADC. The simulations included the effect of the ADC. Each of the radix-2 butterfly stages in Fig. 7 has the structure shown in Fig. 3. Scaling occurs at the input to the block and quantization at the output, except for the first stage in Fig. 7 which has no scaling as this is equivalent to changing the input power and the form of the input ADC. The input signal was modeled as a complex signal with Gaussian real and imaginary components. In practice the form of the received OFDM signal will depend on the channel. The performance for Gaussian inputs gives a good indication of the likely performance for most channels, and the relative performance of different designs of FFT.

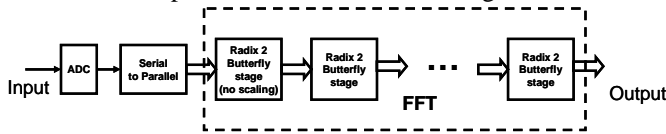


Fig. 7. Model used for evaluation of the receiver FFT.

Simulations were run for different designs of FFT. The MSE of the output from the new technique was calculated in two ways: per sample and per symbol. The *error per sample* was calculated for each complex value at the FFT output for each transform operation. The *error per symbol* was calculated by averaging the MSE per sample for each transform operation. The second method gives a better indication of the performance in an OFDM system because, in most systems, the error coding operates only on a symbol by symbol basis with no interleaving between symbols. Results were considered for four different FFT designs: the new technique using scaling of $1/\sqrt{2}$ at each stage; the new technique using scaling of 1 and 0.5 at alternate stages; a conventional FFT design with scaling of 0.5 at each stage; and an FFT using BFP. The impairment caused by the ADC alone was also calculated. Simulations were performed for a number of values of FFT size, N , and varying number of bits, b , of fixed-point precision.

Fig. 8 shows the MSE per sample for different FFT designs and for the ADC alone using 16 bit precision. The error for the ADC alone falls as the headroom increases from 0dB to approximately 14dB. The results for the FFTs have a generally similar form. The minimum has increased by 3dB, but this is because there is a 3dB rise in power between the input to the ADC and the quantization at the first butterfly. In other words the minimum is also occurring at around 14 dB headroom for the quantization stages. This is consistent with the results in Fig. 6 for quantization of Gaussian variables. To the right of the minimum, most of the graphs slope up with a gradient of unity. The conventional design using scaling of 0.5 at each stage has the worst performance. This is because the signal power is decreasing at each stage. The two versions of the new technique give better performance, with only a small penalty for using the alternating 0.5 and 1 scaling. The performance

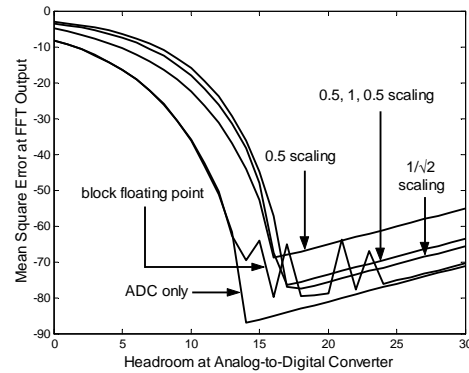


Fig. 8. Normalized MSE per sample versus the headroom at the input to the ADC for varying FFT design, 16 bit precision and $N = 64$.

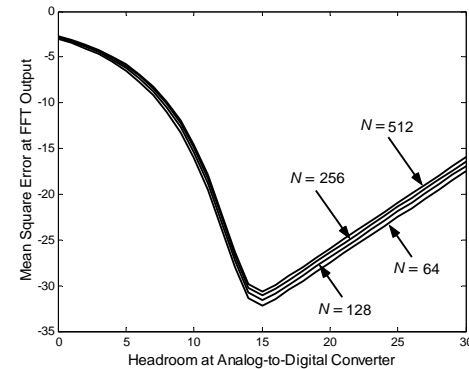


Fig. 9. Normalized MSE per sample versus the headroom at the input to the ADC for $1/\sqrt{2}$ scaling, 8 bit precision and varying N .

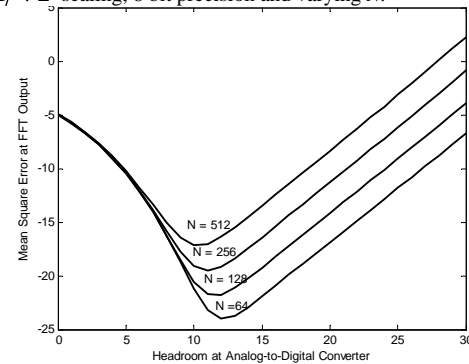


Fig. 10. Normalized MSE per sample versus the headroom at the input to the ADC for 1/2 scaling, 8 bit precision and varying N .

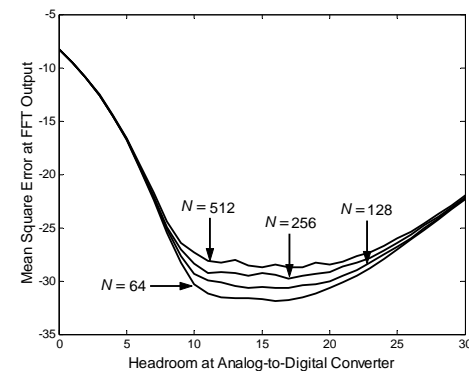


Fig. 11. Normalized MSE per sample versus the headroom at the input to the ADC for block floating point, 8 bit precision and varying N .

for BFP shows very significant changes as the headroom changes slightly. The results for BFP also vary dramatically between simulation runs. This is because the performance is dominated by very low probability ‘bad symbols’. These are symbols with very high peak to average ratio which cause the BFP to scale by 0.5 at every stage to avoid clips. This results in high values of rounding noise. The performance for these bad symbols can change significantly for small changes in headroom. This is because if the high peak is clipped at the ADC, less scaling is required by the block floating point.

Fig. 9, 10 and 11 show how the MSE varies with the number of subcarriers for different designs. Comparing Fig. 9 and 10 it can be seen that for the new design the performance degrades much less with increasing N . For BFP the performance is much less sensitive to headroom as the variable scaling compensates to avoid clipping. Due to space limitations, results for the transmitter FFT are not included in this paper, however, results are quite similar.

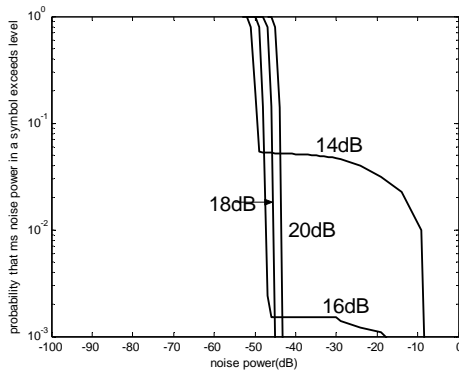


Fig. 12. Complementary cumulative distribution of MSE per symbol for 0.5 weighting, 16 bit fixed-point, $N = 64$ and $h_r = 14, 16, 18$ and 20dB.

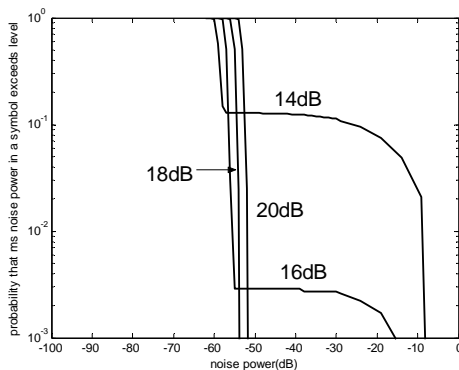


Fig. 13. Complementary cumulative distribution of MSE per symbol for alternating 0.5, 1 weighting, 16 bit fixed-point, $N = 64$ and $h_r = 14, 16, 18$ and 20dB.

In OFDM the per symbol statistics are also important. Figs. 12 and 13 show the CCD for conventional 0.5 weighting and the new method using alternating 0.5 and 1 weighting for values of h_r just below and above the value for MMSE per sample. The parameters are the same as for Fig. 8. In Fig 14, it can be seen that although $h_r = 14$ dB results in a low MSE per symbol for most symbols, there is a plateau around 0.05. In

other words, when $h_r = 14$ dB, around 5% of symbols have higher values of MSE per symbol. When the threshold is increased, the probability of the plateau falls rapidly and for 18 and 20dB is no longer visible on the graph. However because of the infinite nature of the Gaussian distribution, there will always be a plateau. The headroom should be chosen so that the probability associated with this plateau is acceptable. This will usually be slightly above the headroom for MMSE. Fig. 13 shows the results for the new method. Note that the noise power is significantly lower.

VIII. FFT STRUCTURES AND FIXED-POINT IMPLEMENTATION

In order to demonstrate the feasibility of implementing these new scaling methods on a DSP, code was written for an Analog Devices Blackfin[®] processor [8]. This processor features both DSP and MCU functionality. The Blackfin processor used in this case was an ADSP-BF533 which features up to 1,512 MMACs (Mega Multiply Accumulates per second), 1.2Mbits of high speed SRAM and extremely low power consumption.

The code was adapted from the current range of FFT algorithms readily available from the Blackfin website [9]. The implementation is already highly optimised and takes full advantage of the existing Blackfin instruction set support for the FFT. The Analog Devices VisualDSP++ IDE (Integrated Development Environment) [10] was used to implement and debug the code. Matlab was used as the data generation and analysis tool. The FFT function in Matlab was used to provide a reference FFT implementation and also for generating the input noise samples. Matlab scripts were also written to analyse the results.

In this particular case a 256-point radix-2 decimation in time algorithm was chosen for the implementation. The algorithm implements a 16 bit complex FFT. Two scaling methods were compared; the first was the standard scaling method of scaling by one half at every butterfly stage. The second method was scaling by one half at every second butterfly stage.

A test harness was created where Gaussian distributed input data was generated and passed to the DSP implementation via means of an input file. The DSP test harness reads the input data and processes the data through the DSP code producing an output file which is read by the Matlab script and analysed.

The DSP implementation takes advantage of a very useful tool provided in VisualDSP++, called the compiled simulator. The compiled simulator generates an x86 PC based executable which simulates the DSP program to the bit-exact level. This PC executable allows full debug capabilities, while running thousands of times quicker than a standard simulation. The compiled simulator is especially useful in this case since it can also be invoked directly as an executable from Matlab. The compiled DSP executable reads files off the PC hard drive, allowing for simple creation of batch testing and analysis.

In the Matlab analysis, the output data from the DSP is compared with the reference FFT implementation provided in Matlab. The MSE is calculated by comparing these two results. The level is varied across the complete 16 bit range of the input precision and the MSE is plotted against the level.

The results are presented in Fig. 14. All curves in Fig. 14 are the result of processing Gaussian distributed inputs. Note how the performance of the FFT improves as the signal level increases. This is due to the roundoff error becoming less significant as the signal level increases. With standard scaling the best MSE seen is -59.3 dB. With alternate scaling, the best MSE seen is -73.3 dB. This improvement can be attributed to the scaling of the intermediate signals of the FFT reducing the effect of round-off noise while allowing controlled saturation.

The cycles required to implement this FFT on the ADSP-BF533 is 3171. The code size is 500 bytes. The data size is mainly comprised of the twiddle table, and the input and output data arrays totaling to about 3000 bytes for the 256-point complex FFT. To implement the alternate scaling scheme merely required an extra 80 cycles, an extra 100 bytes of code space, and no extra data space.

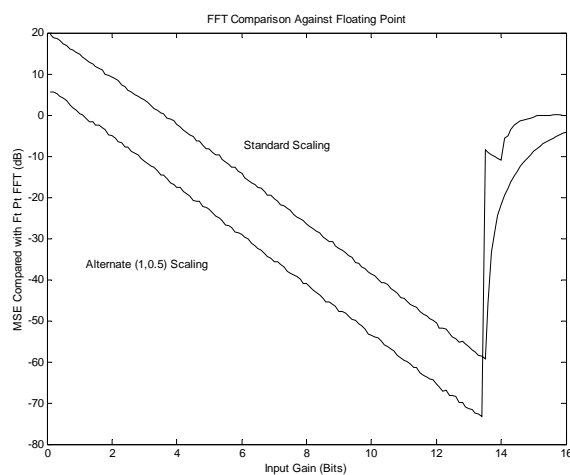


Fig. 14: Comparison of two ADSP-BF533 implementations of a 256-point Radix-2 Complex Decimation in Time FFT.

In conclusion, there are an extremely large number of possible FFT types, sizes and precision possibilities. This particular example showed an improvement of around 15dB in the MSE of the result, with very little cycles overhead. The improvement is greater for larger FFT sizes. This improvement is significant enough that it could either improve the performance of the OFDM implementation, allow a smaller/faster implementation on a DSP, or allow the implementation of silicon-based designs using fewer bits. The improvements in a silicon-based design will come from lower memory requirements and fewer bits required in the arithmetic units. This translates into smaller die area, lower power consumption and reduced costs.

IX. CONCLUSIONS

In this paper we have described a new approach to FFT design which gives significantly improved performance in OFDM transmitters and receivers. FFTs implemented using fixed-point arithmetic, are usually designed so that there is no possibility of numeric overflow. This paper shows that, in an OFDM system, better overall performance can be achieved for a given fixed-point precision, if overflow is allowed to occur with low probability, rather than being completely avoided

(provided that overflow results in saturation rather than wrap-around). Results are presented for the statistics of signals within the FFT of an OFDM transmitter. It is shown that at many points within the FFT structure the signals have an approximately Gaussian distribution. The effect of fixed-point implementation of OFDM is therefore closely related to the quantization of Gaussian variables. Results are presented which demonstrate the distribution of errors when a Gaussian random variable is quantized. It is shown that the distribution of errors depends on the headroom; that is the ratio of the maximum quantization level to the RMS power of the random variable. Graphs of MSE versus headroom show that there is a value of headroom which results in MMSE. However when the distribution of the errors is considered, in addition to the MS value, it is found that the optimum headroom is generally slightly above the MMSE value. The MMSE headroom depends on the resolution. For 16-bit arithmetic it is between 14 and 15 dB.

The new FFTs are designed so that scaling results in the signals having approximately the optimum headroom throughout the FFT. For a radix-2 FFT, this is achieved by scaling by $1/\sqrt{2}$ at each stage. A slightly suboptimal, but more practical implementation is to use scaling factors of 1 and 0.5 at alternate stages. The improvement in SNR which can be gained by using the new technique increases with the size of the FFT. For a 64-point FFT, the SNR of the new technique is about 10dB less than that of an FFT using conventional scaling of 0.5 at each stage. Each doubling in size of the FFT results in an almost 3 dB increase in the advantage of the new technique.

The new technique was implemented on an Analog Devices processor in fixed-point arithmetic and the theoretical results were verified. In the particular example implemented, approximately 15 dB of improvement was achieved. This corresponds to around 2.5bits lower precision needed to achieve the same performance of the FFT with standard scaling. These lower precision requirements translate into lower system cost, lower power consumption, reduced silicon die area and higher system performance.

REFERENCES

- [1] R. Van Nee and R. Prasad, *OFDM for Multimedia Communications*, Artech House, 2000.
- [2] A. V. Oppenheim, R. W. Schaffer and R. W. Buck, *Discrete-Time Signal Processing*. Second Edition, Prentice Hall, 1999.
- [3] Y. Ma, "An accurate error analysis model for fast Fourier transform," *IEEE Trans. Signal Processing*, vol. 45, pp. 1641-1645, June 1997.
- [4] T. Lenart and V. Owall, "A 2048 complex point FFT processor using a novel data scaling approach," in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS '03)*, Bangkok, May 2003, pp. 45-48.
- [5] H. A. Suraweera and J. Armstrong, "Noise bucket effect for impulse noise in OFDM," *IEE Electron. Lett.*, submitted.
- [6] J. Max, "Quantization for minimum distortion," *IRE Trans. Inform. Theory*, vol. IT-6, pp. 7-12, Mar. 1960.
- [7] D. Dardari, "Exact analysis of joint clipping and quantization effects in high speed WLAN receivers," in *Proc. IEEE ICC '03*, Anchorage, AK, USA, May 2003, pp. 3487-3492.
- [8] <http://www.analog.com/processors/processors/blackfin/index.html>
- [9] <http://www.analog.com/processors/processors/blackfin/technicalLibrary/manuals/codeExamples.html>
- [10] VisualDSP++ 3.5 Users Guide for 16-bit Processors, Analog Devices, 2003.