NON-UNIFORM WORDLENGTH DELAY LINES FOR FIR FILTERS

Gregour Bolton, Robert W. Stewart

DSP Enabled Communications Group, Department of Electronic and Electrical Engineering, University of Strathclyde
204 George Street, G1 1XW, Glasgow, Scotland
phone: + (44) 0141 548 2605, email: gregour.bolton@.strath.ac.uk
web: www.eee.strath.ac.uk

ABSTRACT

When FIR filters are designed floating point arithmetic is generally used. However when implemented on hardware such as ASICs, fixed point arithmetic must be used to minimise cost and power requirements. Research to minimise hardware costs has mainly focused on the quantization effects of fixed point wordlengths for the coefficients, multipliers and adders of FIR filters, but with the actual data delays assigned a uniform wordlength and essentially not optimised. This paper proposes that the wordlengths of the delay line can be non-uniform with a minimal increase in quantization noise for parallel implementation of FIR filters where there are differences in the magnitudes of the coefficients. A non-uniform delay line allows hardware savings in terms of delay register wordlengths, delay signal wordlengths and multiplier wordlengths. Results for an FIR design are presented which demonstrate the hardware savings when using a non-uniform wordlength delay line.

1. INTRODUCTION

FIR filters are one of the most commonly used components in DSP systems. When FIR filters are designed floating point arithmetic is commonly used, however when implemented on hardware such as ASICs and FPGAs, fixed point arithmetic is generally used. Fixed point implementation is used to minimise hardware cost and power usage and to maximise performance yet by minimising wordlengths numerical precision is lost. The conversion from floating point to fixed point reduces precision and introduces noise in the form of quantization errors due to rounding or truncation.

Floating point to Fixed point Conversion (FFC) aims to minimise hardware requirements while maintaining the numerical accuracy of the system being converted. Several FFC techniques have been developed some of which are specific to FIR filters [1][2][3] and others that are for more general DSP systems[4][5][6]. FFC algorithms have error constraints which determine if the fixed point implementation is suitable.

In this paper, the quantisation effects of individual delays and delay signals for parallel implementations of FIR filters are examined. A FFC algorithm examined in [6] is used to compare savings in hardware when a non-uniform wordlength delay line is implemented over a uniform one. The results show that savings can be made while remaining within the specified error constraint.

2. BACKGROUND

Fixed point numerical representation uses a series of bits in binary format to represent a value. It is specified in the form of <(*s*),*iwl*,*fwl*> where (*s*) indicates unsigned or twos compliment representation , *iwl* is the integer wordlength including sign bit if applicable, *fwl* is the fractional wordlength and *wl* is the total wordlength as shown in Figure 1.

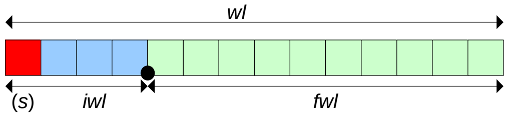

Figure 1- Fixed point specification

The goal of FFC is to minimise the total wordlength of all the fixed point signals in a DSP system while maintaining numerical accuracy. The *iwl* can be calculated by determining the dynamic range of a signal and then assigning the minimal number of bits that ensures that no overflow occurs. The dynamic range can be determined by simulating a floating point implementation of the system and analysing the data at each signal. To minimise hardware requirements fractional bits are truncated as operations such as addition and multiplication increase wordlength in order to maintain accuracy.

Truncation of the *fwl* introduces noise in the form of quantisation errors where there are insufficient bits to represent a value. The quantisation error is the difference between the representable value for a given fixed point representation and the actual value. Signal to Quantisation Noise Ratio (SQNR) is the most common [4][7] measure of quantisation noise in a system being converted using FFC. SQNR is measured by comparing the output of a floating point implementation of a system with the output of a fixed point implementation of the same system

$$SQNR(dB) = 10\log\frac{P_{Signal}}{P_{Noise}} \qquad (1)$$

SQNR is calculated using (1) where the P_{Signal} is the average power of the floating point output data and P_{Noise} is the average power of the difference between the floating point output data and the fixed point output data due to quantisation noise. Using SQNR as constraint, a target SQNR can be set that the FCC must meet or exceed during conversion.

3. NON-UNIFORM WORDLENGTH DELAY LINES

The standard parallel implementation of an FIR filter shown in Figure 2 uses a series of delays {$z_1 \ldots z_N$} to hold input samples *x(k)*. The input samples held in the delays are multiplied by the coefficients {c_0, $c_1 \ldots c_N$} and accumulated by the adders.

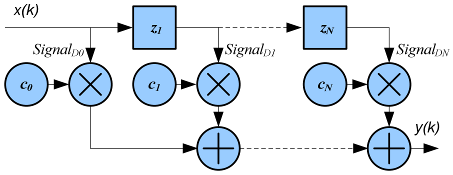

Figure 2-Signal flow graph of standard parallel implementation of an FIR

Also shown in Figure 2 are the signals {$Signal_{D0}$, $Signal_{D1} \ldots Signal_{DN}$} that carry the samples from the delays to the multipliers. When implemented in hardware these signals also have a finite wordlength which introduces quantisation noise. Since these signals are normally assigned the same wordlength as the delays they tend not to add any additional quantisation noise to the output of the FIR.

To illustrate the contribution of quantisation noise from non-uniform wordlength delay signals first consider the FIR filter described by the following difference equation:

$$y(k) = x(k)c_0 + x(k-1)c_1 + \ldots + x(k-N)c_N \tag{2}$$

where c_n denotes the filter coefficients, $y(k)$ is the output signal and $x(k\text{-}n)$ is the input signal delayed by n samples. This difference equation represents the desired signal.

$$\begin{aligned} y(k) + q_T &= (x(k) + q_0)c_0 + (x(k-1) + q_1)c_1 + \\ &+ \ldots + (x(k-N) + q_N)c_N \end{aligned} \tag{3}$$

Delay signal quantisation noise q_n is then added in (3). The noise propagation models from [4] for addition:

$$Z = X + Y \Rightarrow \begin{cases} z = x + y \\ q_z = q_x + q_y \end{cases} \tag{4}$$

and multiplication:

$$Z = X \times Y \Rightarrow \begin{cases} z = xy \\ q_z = q_x y + q_y x + q_x q_y \end{cases} \tag{5}$$

can be used to rearrange the difference equation. In the noise propagation models the output Z is the result of an operation using two inputs X and Y. Z, X and Y represent the signals z, x and y and their respective quantisation noise q_z, q_x and q_y.

$$\begin{aligned} y(k) + q_T &= q_0 c_0 + x(k)c_0 + q_1 c_1 + x(k-1)c_1 + \\ &+ \ldots + q_N c_N + x(k-N)c_N \end{aligned} \tag{6}$$

The desired signal (2) can be removed from the rearranged the difference equation (6) thus leaving only the quantisation noise from the delay signals.

$$q_T = q_0 c_0 + q_1 c_1 + \ldots + q_N c_N \tag{7}$$

In (7) it is shown that the magnitude of the quantisation noise from each of the delay signals is proportional to the magnitude of the coefficient multiplying it. Thus quantisation noise contribution for a delay signal multiplied by a coefficient with a small magnitude will not be as great as the quantisation noise contribution for delay signals multiplied by coefficients with larger magnitude.

Therefore the fractional wordlengths of delay signals can be reduced where they are multiplied by coefficients with small magnitudes while delay signals multiplied by coefficients with larger magnitudes are assigned longer fractional wordlengths. This can be achieved without significantly decreasing the SQNR at the output of the filter.

Reducing the wordlength of delay signals also allows a reduction in the wordlength at the output of the multipliers. In order to maintain numerical accuracy the wordlength at the output of a multiplier should be the sum of the wordlengths at its input. Therefore where the wordlength of a delay signal can be reduced the wordlength of the multiplier it is connected to can be reduced by the same amount without any further loss of numerical accuracy.

4. FIR EXAMPLE

The delay signals of an example FIR are converted to fixed point using the FFC bitless algorithm [6]. The bitless or alternatively named b_{max}-1 algorithm starts with wordlengths of signals to be converted to fixed point at the maximum representable wordlength of the fixed point simulator. The algorithm then temporally decrements by one bit the *fwl* of each signal while the *fwl* of all other signals remain unchanged. The signal that maximises the SQNR is allowed to keep the removed bit. This process is repeated until no further fractional bits can be removed from any signal without falling below the target SQNR.

Starting at the maximum representable wordlength of the fixed point simulator leads to a very large search space therefore the search space is reduced with a binary search algorithm before the bitless algorithm is applied.

For clarity and the purposes of demonstration only the delay signals are converted while all other signals remain floating point values. This ensures that the only noise at the output of the filter is from quantisation of the delay signals. The example FIR is a normalised fifteen coefficient low pass filter with a stop band attenuation of -60dB and a transition

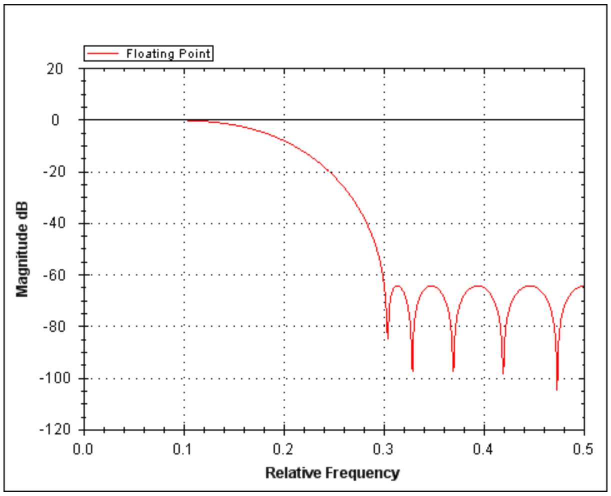

Figure 3-Magnitude Response of Example FIR

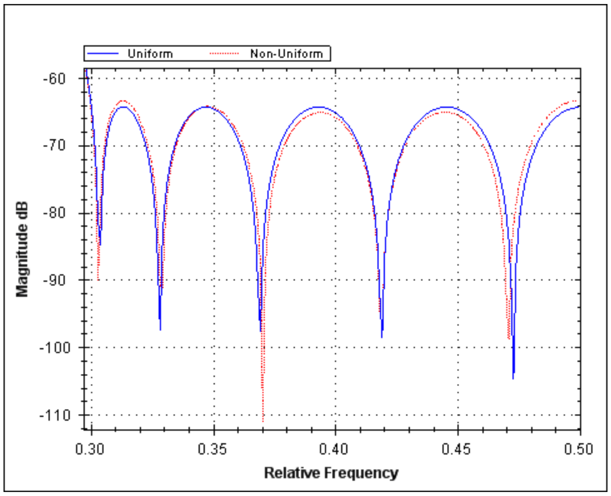

Figure 4-Stop Band Region Magnitude Response of Uniform and Non-Uniform Delay Lines

		Truncation				**Rounding**			
Coefficients		**Uniform**		**Non-Uniform**		**Uniform**		**Non-Uniform**	
Index	**Value**	**Signal$_D$**	**Delay**	**Signal$_D$**	**Delay**	**Signal$_D$**	**Delay**	**Signal$_D$**	**Delay**
0	0.00431622	14	NA	7	NA	13	NA	8	NA
1	0.00740138	14	14	8	14	13	13	8	14
2	-0.01014178	14	14	8	14	13	13	8	14
3	-0.04423428	14	14	10	14	13	13	11	14
4	-0.03032523	14	14	10	14	13	13	10	14
5	0.09640909	14	14	12	14	13	13	12	14
6	0.28454928	14	14	14	14	13	13	14	14
7	0.3770314	14	14	14	14	13	13	14	14
8	0.28454928	14	14	14	14	13	13	14	14
9	0.09640909	14	14	12	12	13	13	12	12
10	-0.03032523	14	14	10	10	13	13	10	10
11	-0.04423428	14	14	10	10	13	13	11	11
12	-0.01014178	14	14	8	8	13	13	8	8
13	0.00740138	14	14	8	8	13	13	8	8
14	0.00431622	14	14	7	7	13	13	8	8
Total		210	196	152	167	195	182	156	169
Est. Saving (%)		NA		27.62	14.8	NA		20	7.14

Table 1-Fractional Wordlength of Delays and Delay Signals

band from (1/5) of the sampling rate to (3/5) of the sampling rate.

Figure 3 shows the magnitude response of the filter using a floating point implementation. During FFC the filter is stimulated by uniformly distributed noise source with a lower bound of -1 and an upper bound of 1. An error constraint of 80dB SQNR between the floating point simulation and fixed point simulation is set. Both rounding and truncation quantisation modes are tested.

5. RESULTS

The results of the FFC using the bitless algorithm are shown in Table 1. The *fwl* of the delay signals (**Signal$_D$**) for both the uniform and non-uniform implementations shown in the right hand columns correspond to the coefficient values that they are multiplied by which are shown in the left hand column.

The *fwl* of the FIR delays (**Delay**) have also been added to the table. The *fwl* of the delays have been calculated by maintaining precision along the delay line as required by the delay signals. As the FIR was stimulated by a uniformly distributed noise source with a lower bound of -1 and an upper bound of 1 all the delay signals are signed and are assigned one integer bit.

The magnitude response of the uniform and non-uniform delay line in truncation mode is shown in Figure 4. As there is minimal difference in the pass band and transition band regions only the stop band is shown. While the non-uniform response differs from the uniform response the basic shape is intact and both implementations are below the -60dB stop band attenuation of the filter specification. Experiments have shown that as the FFC SQNR target is reduced the magnitude response of the non-uniform implementation becomes increasingly distorted when compared to the uniform delay line which remains very close to the floating point implementation magnitude response.

The estimated saving is based on the total number of fractional bits used for uniform against non-uniform implementations. There is a 14.8% saving of delay register wordlengths for truncation and 7.14% for rounding. In devices such as FPGAs reducing the wordlengths of the delay registers reduces the number of flip flops used by the synthesized implementation of the filter. The rounding mode has a smaller saving due to the uniform delay line implementation using a shorter *fwl* for rounding than truncation.

6. CONCLUSION

This paper proposes the assignment of a non-uniform wordlength delay line for parallel implementation FIR filters. The proposed implementation saves hardware in terms of delay register wordlengths, delay signal wordlengths and multiplier wordlengths.

This implementation has been shown to reduce hardware requirements while maintaining filter specification characteristics and numerical accuracy. This has been demonstrated for a fifteen coefficient low pass filter with a parallel implementation using the bitless FFC algorithm. In addition this implementation can be applied to high pass, band pass and band stop filters.

REFERENCES

[1] J. Qiao, P. Fu, and S. Meng, "A Combined Optimization Method of Finite Wordlength FIR Filters", in *Proc. First International Conference on Innovative Computing, Information and Control*, Beijing, China, August 30-September 1. 2006, pp. 103–106.

[2] R.V. Kacelenga, P.J. Graumann and L.E. Turner, "Design of digital filters using simulated annealing", in *Proc. of the IEEE ISCAS'90, International Symposium on Circuits and Systems*, New Orleans, U.S.A., May 1-3. 1990, pp. 642–645.

[3] D.J. Xu and M.L. Daley, "Design of finite word length FIR digital filter using a parallel genetic algorithm", in *Proc. of the 1992 IEEE Southeastcon*, Birmingham, AL, U.S.A., April 12-15. 1992, pp. 834–837.

[4] D. Menard, R. Rocher and O. Sentieys, "Analytical Fixed-Point Accuracy Evaluation in Linear Time-Invariant Systems", *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 55, issue 10, pp. 3197–3208, Nov. 2008.

[5] C. Shi and R.W. Brodersen, "Automated fixed-point data-type optimization tool for signal processing and communication systems", in *Proc. 41st Design Automation Conference*, San Diego, U.S.A., June 7-11. 2004, pp. 478–483.

[6] M.-A. Cantin, Y. Savaria and P. Lavoie, "A comparison of automatic word length optimization procedures", in *Proc. IEEE ISCAS'02, International Symposium on Circuits and Systems Volume 2*, Scottsdale, Arizona, U.S.A., May 26-29. 2002, pp. 612–615.

[7] W. Sung and K. Kum, "Simulation-based word-length optimization method for fixed-point digital signal processing systems", *IEEE Transactions on Signal Processing*, vol. 43, issue 12, pp. 3087–3090, Dec. 1995.