PAPER

Low-Complexity Constant Multiplication Based on Trigonometric Identities with Applications to FFTs

Fahad QURESHI[†a)], ***Nonmember and*** **Oscar GUSTAFSSON**[†b)], ***Member***

SUMMARY In this work we consider optimized twiddle factor multipliers based on shift-and-add-multiplication. We propose a low-complexity structure for twiddle factors with a resolution of 32 points. Furthermore, we propose a slightly modified version of a previously reported multiplier for a resolution of 16 points with lower round-off noise. For completeness we also include results on optimal coefficients for eight points resolution. We perform finite word length analysis for both coefficients and round-off errors and derive optimized coefficients with a minimum complexity for varying requirements.

key words: *Complex multiplier, FFT, Constant multiplication, Shift-and-add multiplication*

1. Introduction

Computation of the discrete Fourier transform (DFT) and inverse DFT is used in e.g. orthogonal frequency-division multiplexing (OFDM) communication systems and spectrometers. An N-point DFT can be expressed as

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk}, k = 0, 1, N-1, \quad (1)$$

where $W_N = e^{-j\frac{2\pi}{N}}$ is twiddle factor, the N:th primitive root of unity with its exponent being evaluated modulo N, n is the time index, and k is the frequency index. Various methods for efficiently computing (1) have been the subject of a large body of published literature. These methods are commonly referred to as fast Fourier transform (FFT) algorithms. Also, many different architectures to efficiently map the FFT algorithm to hardware have been proposed [1].

A commonly used architecture for transforms of length $N = b^r$ is the pipelined FFT [2–7]. The pipeline architecture is characterized by continuous processing of input data. In addition, the pipeline architecture is highly regular, making it straightforward to automatically generate FFTs of various lengths.

Figure 1 outlines the architecture of a Radix-2^i, single-path delay feedback (SDF), pipelined FFT architecture for $N = 256$. This architecture is generic while the required ranges of each complex twiddle factor multiplier for different algorithm are outlined in Table 1 [5–8].

Table 1 Multiplication at different stages for different architecture ($N = 256$).

	Stage number						
Radix	1	2	3	4	5	6	7
2	W_{256}	W_{128}	W_{64}	W_{32}	W_{16}	W_8	W_4
2^2 [5]	W_4	W_{256}	W_4	W_{64}	W_4	W_{16}	W_4
2^3 [6]	W_4	W_8	W_{256}	W_4	W_8	W_{32}	W_4
2^4 [7]	W_4	W_8	W_{16}	W_{256}	W_4	W_8	W_{16}
2^5 [8]	W_1	W_8	W_{16}	W_{32}	W_{256}	W_1	W_8
M. 2^4 [7]	W_4	W_{16}	W_4	W_{256}	W_4	W_{16}	W_4

We will from now on denote a multiplier with a twiddle factor resolution of N points around the unit circle a W_N-multiplier. For small ranges of the twiddle factor multipliers it is advantageous to use arithmetic circuits optimized for the required coefficients rather than general multipliers. A W_4 multiplier only performs multiplication by one of $\{1, j, -1, -j\}$, in practice 1 or $-j$, and is most commonly realized by combining it in the subsequent butterfly (often denoted BFII as opposed to the standard butterfly BFI following [5]). For larger N it is common to utilize the octave symmetry of the coefficients. This means that only twiddle factors corresponding to angles in the range $0 \leq \alpha \leq \pi/4$ needs to be considered. This is equivalent to twiddle factors in the range $0 \leq m \leq N/8$. Multiplications for other values of m can be obtained by optionally swapping outputs (symmetry around $\Re(m) = \Im(m)$ and $\Re(m) = \Im(m)$) and negating one or both outputs (symmetry around $\Re(m) = 0$ and $\Im(m) = 0$). A complete complex multiplier based on complex constant multiplication is shown in Fig. 2. In previous work, the complex multipliers for W_8 and W_{16} have been replaced with constant complex multipliers based on shift-and-add networks for performing the multiplication [10–12].

In this work, we propose a low complexity complex constant W_{32}-multiplier based on trigonometric identities. Furthermore, we revisit and slightly improve the W_{16}-multiplier proposed in [7] and, for completeness, calculate the coefficients and complexity of the W_8-multiplier. A preliminary version of this work was presented in [9]. In this extended version, we have included results on the finite word length properties. It is shown that the coefficient quantization can lead to

Manuscript received July 07, 2010.

[†]The authors are with the Department of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden. http://www.es.isy.liu.se/

a) E-mail: fahadq@isy.liu.se

b) E-mail: oscarg@isy.liu.se

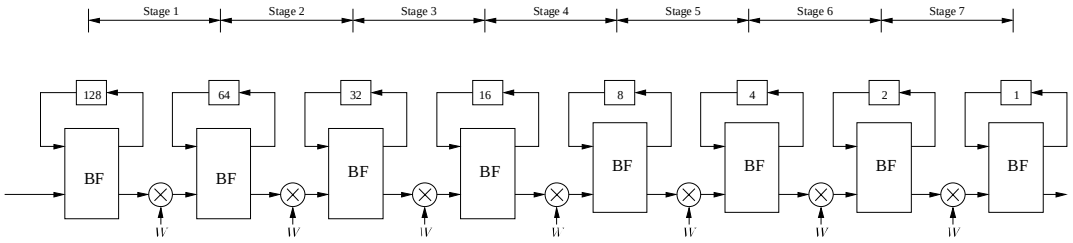


Fig. 1 Radix-2^i single-path delay feedback (SDF) pipeline FFT architecture ($N = 256$) with twiddle factor stages as used in Table 1.

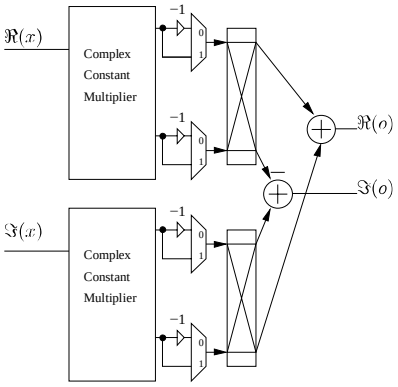


Fig. 2 Block diagram of complex multiplier based on complex constant multipliers.

larger errors than considered in [7, 9]. Furthermore, expressions for the round-off noise are derived. Based on this, a slight modification is proposed for the W_{16}-multiplier from [7].

The rest of the paper is arranged as follows. In the next section, the complex constant multipliers are introduced. Then, in Section 3, coefficient quantization and data round off errors are analyzed. Then, in Section 4, results are presented and finally, some conclusions are given in Section 5.

2. Complex Constant Multipliers

2.1 W_8-Multiplier

For a W_8-multiplier, only a multiplication by either 1 or $\sin\frac{\pi}{4}$ ($\cos\frac{\pi}{4}$) is required. This can easily be realized using a multiplexer selecting between the input or the output of a constant multiplier with coefficient $\sin\frac{\pi}{4}$. The constant multiplier can be realized using a minimum number of adders using the method in [14].

2.2 W_{16}-Multiplier

In [7], a W_{16}-multiplier based on the trigonometric identity

$$\sin 2\theta = 2\sin\theta\cos\theta, \tag{2}$$

was introduced. Hence, as $2\frac{\pi}{8} = \frac{\pi}{4}$ it is possible to

Table 2 Trigonometric identities used for W_{16}-multiplier.

Coefficient	Used expression
$\sin\frac{\pi}{4}$	$2\sin\frac{\pi}{8}\cos\frac{\pi}{8}$
$\sin\frac{\pi}{8}$	$\sin\frac{\pi}{8}$
$\cos\frac{\pi}{8}$	$\cos\frac{\pi}{8}$

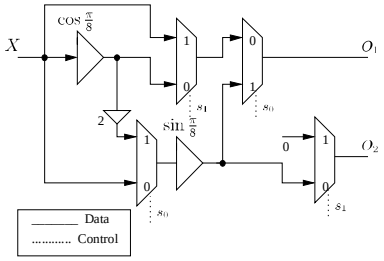


Fig. 3 Complex constant W_{16}-multiplier modified form [7].

compute all the three required values for a W_{16}- multiplier using only two multipliers with the constant values $\sin\frac{\pi}{8}$ and $\cos\frac{\pi}{8}$ as shown in Table 2. The resulting structure is shown in Fig. 3. Note that multiplication by two is equivalent to a left-shift, and, hence, is not considered a multiplication. The structure shown in Fig. 3 is slightly modified compared to that in [7]: two multiplexers are added at the output to allow multiplication by 1 and also the constant coefficient interchange to reduce the round off noise in the structure. Furthermore, it was in [7] suggested that the multipliers should be implemented based on the canonic signed-digit (CSD) representation. In the current work it is instead suggested to use minimum adder multipliers from [14].

2.3 W_{32}-Multiplier

For the W_{32}- multiplier, we propose to use a similar approach, i.e., based on trigonometric identities identify a small number of constant multiplications that can be combined to form all the remaining coefficients. In our proposed approach these constant multiplications are $\sin\frac{\pi}{16}$, $\cos\frac{\pi}{16}$, and $\cos\frac{\pi}{8}$ which can be combined as shown in Table 3. One possible structure for the resulting complex constant multiplier is illustrated in Fig. 4 with the corresponding control signals shown in Table 4. The proposed architecture will be imple-

Table 3 Trigonometric identities used for W_{32}-multiplier.

Coefficient	Used expression
$\sin\frac{\pi}{4}$	$4\cos\frac{\pi}{8}\cos\frac{\pi}{16}\sin\frac{\pi}{16}$
$\sin\frac{\pi}{8}$	$2\cos\frac{\pi}{16}\sin\frac{\pi}{16}$
$\cos\frac{\pi}{8}$	$\cos\frac{\pi}{8}$
$\sin\frac{3\pi}{16}$	$\sin\frac{\pi}{16}\left(2\cos\frac{\pi}{8}+1\right)$
$\cos\frac{3\pi}{16}$	$\cos\frac{\pi}{16}\left(2\cos\frac{\pi}{8}-1\right)$
$\sin\frac{\pi}{16}$	$\sin\frac{\pi}{16}$
$\cos\frac{\pi}{16}$	$\cos\frac{\pi}{16}$

mented by constant multiplication, multiplexers and adders (subtracters).

Table 4 Control signals to obtain the different coefficients for the proposed complex constant W_{32}-multiplier in Fig. 4.

s_0	s_1	s_2	s_3	s_4	O_1	O_2
1	x	x	0	1	0	1
0	0	0	1	1	$\cos\frac{\pi}{16}$	$\sin\frac{\pi}{16}$
1	0	0	1	1	$\cos\frac{\pi}{8}$	$\sin\frac{\pi}{8}$
0	1	1	1	1	$\cos\frac{3\pi}{16}$	$\sin\frac{3\pi}{16}$
1	0	1	1	0	$\cos\frac{\pi}{4}$	$\sin\frac{\pi}{4}$

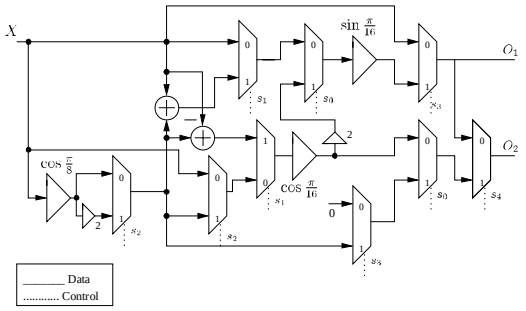


Fig. 4 Proposed complex constant W_{32}-multiplier.

3. Finite word length error analysis

As the proposed structures are based on combinations of several multiplications it is of interest to consider the errors due to coefficient and data quantization. From an FFT point of view, the coefficient quantization will lead to a static deviation from the ideal DFT response. while the data quantization can be seen as a noise source affecting the data. Here, we will consider the absolute magnitude error of the coefficients.

3.1 Coefficient quantization error

We can represent the coefficient quantization error for coefficient c, with quantized value c_q, as Δ_c where

$$c_q = c + \Delta_c. \tag{3}$$

Now, if we use rounding with B fractional bits, we know that $|\Delta_c| \leq 2^{-(B+1)}$. However, given that we know the exact values of both c and c_q, as well as how these errors are propagated one can make a more detailed analysis. Consider the computation of $\sin\frac{\pi}{4}$ in Fig. 4. We have

$$\begin{aligned}\sin\frac{\pi}{4} &= 2\sin\frac{\pi}{8}\cos\frac{\pi}{8} &(4)\\ &= 2\left(\sin\frac{\pi}{8}+\Delta_{\sin\frac{\pi}{8}}\right)\left(\cos\frac{\pi}{8}+\Delta_{\cos\frac{\pi}{8}}\right) &(5)\\ &\approx 2\sin\frac{\pi}{8}\cos\frac{\pi}{8}+2\sin\frac{\pi}{8}\Delta_{\cos\frac{\pi}{8}}\\ &\quad +2\cos\frac{\pi}{8}\Delta_{\sin\frac{\pi}{8}}. &(6)\end{aligned}$$

Where we in (6) consider the first order error terms. Summarizing these errors for the W_{16} multiplier we get the error expressions presented in Table 5. The actual errors using rounding for the partial coefficients are shown in Fig. 5 which shows the relative error in ulps†.

It is clear from this figure that for 7 out of the 16 considered word lengths the magnitude of the error is larger than 0.5 ulp which breaks the precision requirement of the $\sin\frac{\pi}{4}$ multiplication. Conventionally, the word length will increased by one or more bits to achieve required precision. Specially, for the above cases increasing the word length by one bit for one or both of the partial coefficients to meet the specification of all except for the 6 and 12 fractional bits cases. In these cases, the word length must be increased by the two bits to fulfill the requirement of the precision. However, the need for this should be considered on the

Table 5 Coefficient quantization errors for considered multiplier.

Coefficient	First-order expression
$\sin\frac{\pi}{4}$	$2\sin\frac{\pi}{8}\Delta_{\cos\frac{\pi}{8}}+2\cos\frac{\pi}{8}\Delta_{\sin\frac{\pi}{8}}$
$\sin\frac{\pi}{8}$	$\Delta_{\sin\frac{\pi}{8}}$
$\cos\frac{\pi}{8}$	$\Delta_{\cos\frac{\pi}{8}}$

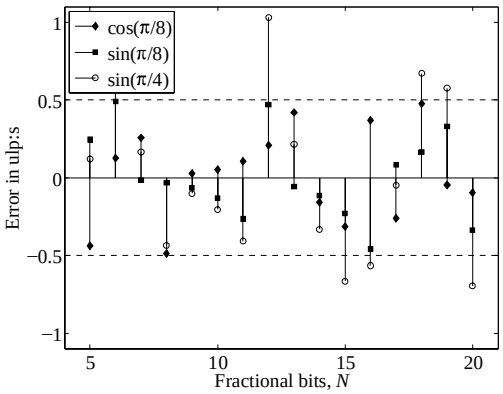


Fig. 5 Relative quantization errors for the coefficients in W_{16}-multiplier using uniform word lengths.

†Unit of least position, i.e., the weight of the least significant bit of the representation. When using B fractional bits, $ulp = 2^{-B}$.

Table 6 Coefficient quantization errors for proposed W_{32}-multiplier.

Coefficient	First-order expression
$\sin\frac{\pi}{4}$	$4\cos\frac{\pi}{16}\sin\frac{\pi}{16}\Delta_{\cos\frac{\pi}{8}} + 4\cos\frac{\pi}{8}\sin\frac{\pi}{16}\Delta_{\cos\frac{\pi}{16}}$ $+4\cos\frac{\pi}{8}\cos\frac{\pi}{16}\Delta_{\sin\frac{\pi}{16}}$
$\sin\frac{\pi}{8}$	$2\sin\frac{\pi}{16}\Delta_{\cos\frac{\pi}{16}} + 2\cos\frac{\pi}{16}\Delta_{\sin\frac{\pi}{16}}$
$\cos\frac{\pi}{8}$	$\Delta_{\cos\frac{\pi}{8}}$
$\sin\frac{3\pi}{16}$	$2\sin\frac{\pi}{16}\Delta_{\cos\frac{\pi}{8}} + \left(2\cos\frac{\pi}{8}+1\right)\Delta_{\sin\frac{\pi}{16}}$
$\cos\frac{3\pi}{16}$	$2\cos\frac{\pi}{16}\Delta_{\cos\frac{\pi}{8}} + \left(2\cos\frac{\pi}{8}-1\right)\Delta_{\cos\frac{\pi}{16}}$
$\sin\frac{\pi}{16}$	$\Delta_{\sin\frac{\pi}{16}}$
$\cos\frac{\pi}{16}$	$\Delta_{\cos\frac{\pi}{16}}$

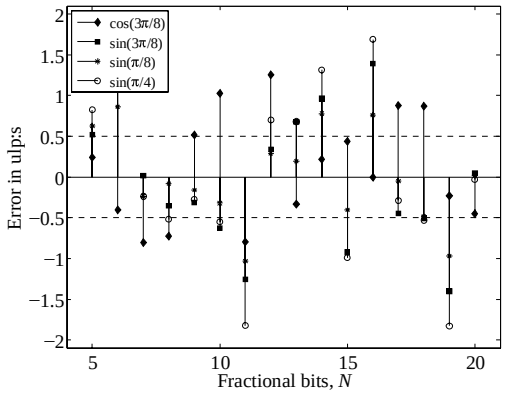


Fig. 6 Relative quantization errors for the derived coefficients in W_{32}-multiplier using uniform word lengths.

system level by evaluating the effect of these additional quantization errors. Note that the magnitudes of the errors for the used partial coefficients are smaller than 0.5 ulp, which is expected as these are derived directly by rounding.

Similarly, error expressions are computed for the different coefficients of the W_{32}-multiplier based on the architecture in Fig. 4. The error expressions of the W_{32}-multiplier is tabulated in Table 6 based on these expressions and the error for varying word lengths is shown in Fig. 6 for those coefficients that are using more than one constant multiplication. Figure 6 shows that, except for 20-bits resolution, at least one of the derived coefficient breaks the precision requirement.

3.2 Round-off noise

In fixed point representation, it is infeasible to increase the word length after each intermediate multiplication stage, product result must be quantized to W-bits. When it comes to data quantization errors this is often modeled as a random noise source with statistical properties, determined by the quantization model and word length. In the proposed architecture one will typically quantize the data after each partial multiplication which is shown in Figs. 7 and 8 for the W_{16} and W_{32}-multipliers, respectively. The resulting noise transfer functions at the output is derived and the results are presented in Tables 7 and 8 for the W_{16} and W_{32}-multipliers, respectively.

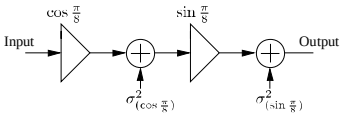


Fig. 7 Output quantization error after $\sin\frac{\pi}{4}$ and $\cos\frac{\pi}{4}$ in W_{16}-multiplier.

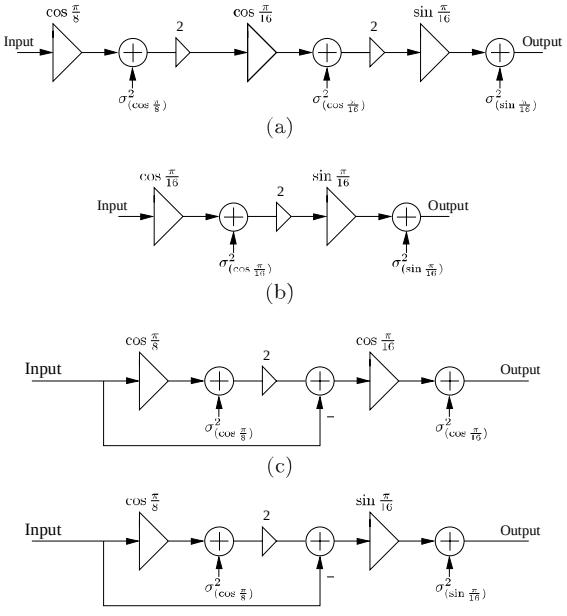


Fig. 8 Output quantization errors in W_{32}-multiplier: (a) $\sin\frac{\pi}{4}$ and $\cos\frac{\pi}{4}$, (b) $\sin\frac{\pi}{8}$, (c) $\cos\frac{3\pi}{16}$, and (d) $\sin\frac{3\pi}{16}$.

Table 7 Noise terms at the W_{16}-multiplier output.

Coefficient	Noise term
$\sin\frac{\pi}{4}$	$\sigma^2_{\cos\frac{\pi}{8}}\left(2\sin\frac{\pi}{8}\right)^2 + \sigma^2_{\cos\frac{\pi}{8}}$
$\sin\frac{\pi}{8}$	$\sigma^2_{\sin\frac{\pi}{8}}$
$\cos\frac{\pi}{8}$	$\sigma^2_{\cos\frac{\pi}{8}}$

Table 8 Noise terms at the W_{32}-multiplier output.

Coefficient	Noise term
$\sin\frac{\pi}{4}$	$\sigma^2_{\cos\frac{\pi}{8}}\left(4\cos\frac{\pi}{16}\sin\frac{\pi}{16}\right)^2 + \sigma^2_{\cos\frac{\pi}{16}}\left(2\sin\frac{\pi}{16}\right)^2$ $+\sigma^2_{\sin\frac{\pi}{16}}$
$\sin\frac{\pi}{8}$	$\sigma^2_{\cos\frac{\pi}{8}}\left(2\sin\frac{\pi}{16}\right)^2 + \sigma^2_{\sin\frac{\pi}{16}}$
$\cos\frac{\pi}{8}$	$\sigma^2_{\cos\frac{\pi}{8}}$
$\sin\frac{3\pi}{16}$	$\sigma^2_{\cos\frac{\pi}{8}}\left(2\sin\frac{\pi}{16}\right)^2 + \sigma^2_{\sin\frac{\pi}{16}}$
$\cos\frac{3\pi}{16}$	$\sigma^2_{\cos\frac{\pi}{8}}\left(2\cos\frac{\pi}{16}\right)^2 + \sigma^2_{\cos\frac{\pi}{16}}$
$\sin\frac{\pi}{16}$	$\sigma^2_{\sin\frac{\pi}{16}}$
$\cos\frac{\pi}{16}$	$\sigma^2_{\cos\frac{\pi}{16}}$

In the original W_{16}-multiplier introduced in [7], the round-off noise term for the $\sin\frac{\pi}{4}$ was

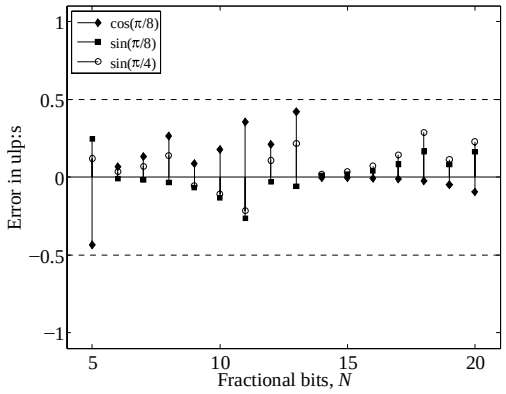


Fig. 9 Relative quantization errors for the coefficients in W_{16}-multiplier using addition aware quantization.

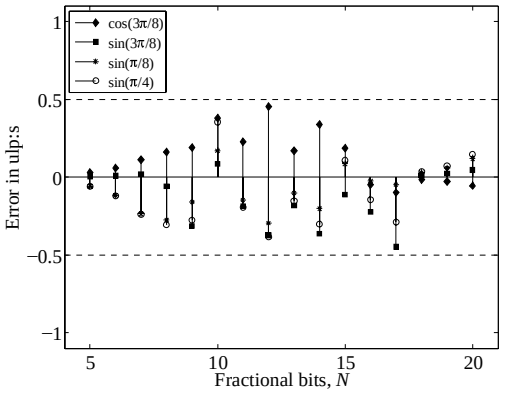


Fig. 10 Relative quantization errors for the coefficients in W_{32}-multiplier using addition aware quantization.

$$\sigma^2_{\sin\frac{\pi}{8}}\left(2\cos\frac{\pi}{8}\right)^2+\sigma^2_{\cos\frac{\pi}{8}}. \tag{7}$$

Compared to the proposed modified W_{16}-multiplier, the round-off noise is reduced corresponding to about one bit lower data word length.

4. Results

4.1 Coefficient quantization and optimized coefficient

As discussed previously, the W_{16} and W_{32}-multipliers are composed of several constant multiplications. Then, the coefficient quantization error of the individual multiplications are combined. While this may lead to cancellation of quantization errors having opposite signs, it may also lead to that the total coefficient quantization error is larger than the individual coefficient quantization errors. The straightforward way of handling this is to increase the word lengths of the individual multiplications until the total error meets the specification.

Addition aware quantization [13] provides a better way of obtaining this increase in accuracy of coefficients. In [13], E additional fractional bits is used to realize that there are exactly 2^E different representable coefficients for which $\epsilon \leq 2^{-(N+1)}$, including the one obtained by rounding to N fractional bits. These 2^E combinations are searched for the best solution.

For each precision requirement, the solution with smallest maximum quantization error among those solutions with the smallest addition count is selected. Here, the coefficient quantization errors of the W_{16} and W_{32}-multipliers are shown in Figs. 9 and 10, respectively. In Fig. 9, it can be seen that in all 7 out of 16 which were breaking the precision requirements in the rounded version are now meeting the precision requirements. In the W_{32}-multiplier, 15 out of 16 cases which was breaking the precision requirement point, now are within the precision requirement.

For a W_8-multiplier implemented with constant coefficient, the optimal coefficients have been tabulated with fractional bits range and correct bits in Table 9. Corresponding results for W_{16} and W_{32}-multipliers are tabulated in Tables 10 and 11, respectively.

Table 9 Coefficient of W_8-multiplier.

Fractional bits	Coefficient $\cos\frac{\pi}{8}$	Addition count	Correct bits
5 – 6	$\frac{45}{64}$	2	6.972
7 – 12	$\frac{2895}{4096}$	3	12.692
13 – 19	$\frac{370733}{524288}$	4	19.322
20	$\frac{741455}{1048576}$	5	20.912

Table 10 Coefficient of W_{16}-multiplier.

Fractional bits	Coefficient		Correct bits
	$\cos\frac{\pi}{8}$	$\sin\frac{\pi}{8}$	
5	$\frac{31}{32}$	$\frac{12}{32}$	5.198
6 – 8	$\frac{945}{1024}$	$\frac{99}{256}$	8.926
9 – 11	$\frac{7567}{8192}$	$\frac{784}{2048}$	11.493
12 – 13	$\frac{7567}{8192}$	$\frac{3135}{8192}$	13.247
14 – 18	$\frac{968753}{1048576}$	$\frac{100319}{262144}$	18.796
19 – 20	$\frac{968753}{1048576}$	$\frac{1605089}{4194304}$	20.931

The hardware resources comparison of the straight forward approach and addition aware method in terms of required number of additions are shown in Figs. 11 and 12 for W_{16} and W_{32} multipliers, respectively. It can be seen that in rare cases the addition aware method can even decrease the number of additions, as can be seen for eight bits.

4.2 Comparison with previous method

Here, a comparison with the previously proposed methods in [10, 11] are presented. The reduced Booth-like multipliers in [11] are based on the observation that when the set of coefficients is known, the Booth-

Table 11 Coefficient of W_{32}-multiplier.

Fractional bits	Coefficient $\cos\frac{\pi}{8}$	Coefficient $\sin\frac{\pi}{16}$	Coefficient $\cos\frac{\pi}{16}$	Correct bits
5 – 7	$\frac{119}{128}$	$\frac{25}{128}$	$\frac{127}{128}$	7.122
8	$\frac{945}{1024}$	$\frac{49}{256}$	$\frac{1007}{1024}$	8.255
9	$\frac{473}{512}$	$\frac{99}{512}$	$\frac{503}{512}$	9.673
10	$\frac{945}{1024}$	$\frac{799}{4096}$	$\frac{1005}{1024}$	10.403
11 – 12	$\frac{3781}{4096}$	$\frac{3197}{16394}$	$\frac{16069}{16384}$	12.144
13 – 14	$\frac{60547}{65536}$	$\frac{12783}{65536}$	$\frac{16069}{16384}$	14.461
15	$\frac{121095}{131072}$	$\frac{25571}{131072}$	$\frac{32133}{32768}$	15.426
16 – 17	$\frac{121095}{131072}$	$\frac{25571}{131072}$	$\frac{128553}{131072}$	17.129
18 – 20	$\frac{968757}{1048576}$	$\frac{204567}{1048576}$	$\frac{4113711}{4194304}$	20.870

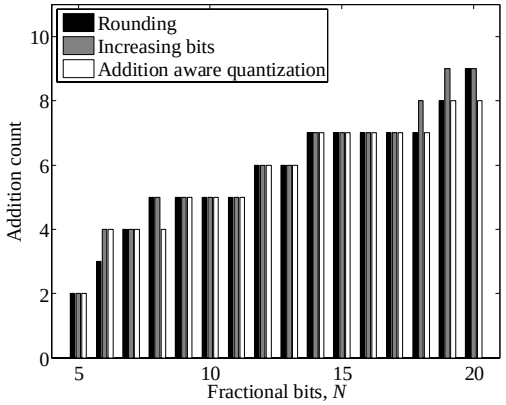


Fig. 11 Addition counts for W_{16}-Multiplier.

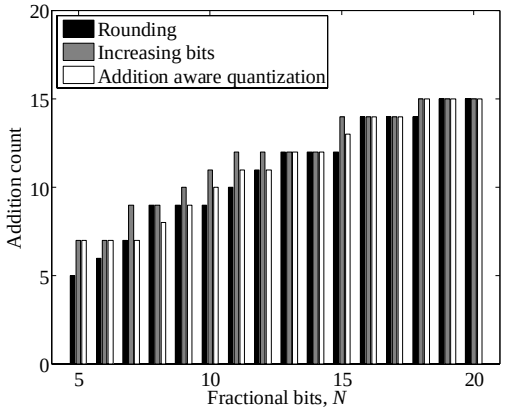


Fig. 12 Addition counts for W_{32}-Multiplier.

encoding logic can be simplified as well as the partial product accumulation tree. Here, we have assumed that four multiplexers are required for each non-zero position in the accumulation tree. This will in practice for some positions be higher. To use multiple constant multiplication (MCM) and a multiplexer to select the correct coefficient was proposed in [10]. For the results presented here, the algorithm in [16] is used, which in general should provide better results compared to the algorithm used in [10].

Table 12 Complexity of W_{16}-multipliers.

Fractional bits	Considered (Fig. 3) Adders[a]	Adders[b]	MUXs	Red. Booth [11] Adders	MUXs	MCM [10] Adders	MUXs
9	5	5	4	8	20	5	4
10	5	5	4	8	20	5	4
11	5	5	4	8	20	5	4
12	6	6	4	8	20	7	4
13	6	6	4	10	24	7	4
14	7	7	4	10	24	8	4
15	7	7	4	10	24	8	4
16	7	8	4	12	28	9	4
17	7	9	4	12	28	10	4
18	7	10	4	12	28	10	4
19	8	10	4	12	28	10	4
20	8	11	4	14	32	12	4

[a]Using minimum adder multipliers from [14].
[b]Using CSD-multipliers as proposed in [7].

The complexity results of the W_{16}-multiplier are shown in Table 12 for a varying number of fractional bits. It is clear that using minimum adder multipliers † [14] is better than CSD multipliers, which is not surprising since CSD multipliers is a subset of minimum adder multipliers. Compared to the reduced Booth-like multipliers, the considered multipliers always have a lower complexity, both in terms of adders and multiplexers. Finally, the MCM approach is as good or slightly worse compared to the complex constant multiplier in Fig. 3.

When it comes to the W_{32}-multipliers, the results are shown in Table 13. Here it can be seen that the adder complexity is typically slightly smaller for the reduced Booth multipliers proposed in [11] compared to the proposed complex constant multiplier in Fig. 4. However, the number of multiplexers is higher in all cases and in most technologies this should mean that the proposed complex constant multiplier has a lower total complexity. Compared to the MCM approach the proposed multiplier has fewer or as few adders, except for the case with nine fractional bits. The advantage of the proposed multiplier increases as the word length increases.

5. Conclusions

In this work, the design of reconfigurable complex constant multipliers was considered, with the focus of rotators in fast Fourier transforms. A multiplier for 32-point resolution was introduced. In addition, a slightly modified previously proposed multiplier for 16-point resolution was discussed. For these two multipliers, finite word length properties for both data and coefficient quantization was discussed and optimal coefficients were derived. For completeness, the optimal co-

†For coefficients longer than 19 bits, the heuristic in [15] is used.

Table 13 Complexity of W_{32}-multipliers.

Fractional bits	Proposed (Fig. 4)		Red. Booth [11]		MCM [10]	
	Adders	MUXs	Adders	MUXs	Adders	MUXs
9	9	9	8	20	8	8
10	10	9	9	22	9	8
11	11	9	9	22	10	8
12	11	9	9	22	11	8
13	12	9	11	26	13	8
14	12	9	11	26	15	8
15	13	9	11	26	15	8
16	14	9	13	30	15	8
17	14	9	13	30	18	8
18	15	9	14	32	19	8
19	15	9	14	32	18	8
20	15	9	16	36	22	8

efficients for multipliers with eight points resolution was also derived.

The results show that the proposed 32-point multiplier has lower complexity compared to earlier work. Also, the 16-point multiplier was compared with earlier work and was shown to have low complexity. Furthermore, the proposed modification leads to slightly lower round-off noise.

6. Acknowledgments

F. Qureshi was supported by the Higher Education Commission, Pakistan. O. Gustafsson was supported by a career contract from Linköping University, Sweden.

References

[1] L. Wanhammar, DSP Integrated Circuits, Academic Press, 1999.

[2] E. H. Wold, A. M. Despain, "Pipeline and parallel-pipeline FFT processors for VLSI implementations," IEEE Trans. Comp., vol.33, no.5, pp.414–426, May 1984.

[3] L. Yang, K. Zhang, H. Liu, J. Huang, and S. Huang. "An efficient locally pipelined FFT processor," IEEE Trans. Circuits Systems II: Express Briefs, vol.53, no.7, pp.585–589, July 2006.

[4] S. Lee, Y. Jung and J. Kim, "Low complexity pipeline FFT processor for MIMO-OFDM systems," IEICE Electronics Express, vol.4, no.23, pp.750–754, December 2007.

[5] S. He and M. Torkelson, "A new approach to pipeline FFT processor," in Proc. IEEE Parallel Processing Symp., 1996, pp.766–770.

[6] S. He and M. Torkelson, "Designing pipeline FFT processor for OFDM(de)Modulation," in Proc. IEEE URSI Int. Symp. Sig. Elect., 1998. pp.257–262.

[7] J.-E. Oh and M.-S. Lim, "New radix-2 to the 4th power pipeline FFT processor," IEICE Trans. Electron., vol.E88-C, no.8, pp.694–697, August 2005.

[8] A. Cortes, I. Velez and J. F. Sevillano, "Radix r^k FFTs: matricial representation and SDC/SDF pipeline implementation," IEEE Trans. Signal Processing, vol.57, no.7, pp.2824–2839, Jul. 2009.

[9] F. Qureshi and O. Gustafsson, "Low-complexity reconfigurable complex constant multiplication for FFTs," Proc. IEEE Int. Symp. Circuits Systems, Taipei, Taiwan, May 2009, pp.1137–1140.

[10] W. Han, T. Arslan, A. T. Erdogan and M. Hasan, "High-performance low-power FFT cores," ETRI J., vol.30, no.3, pp.451–460, June 2008.

[11] Y.-E. Kim, K.-J. Cho, and J.-G. Chung, "Low power small area modified booth multiplier design for predetermined coefficients,"IEICE Trans. Fund., vol.E90-A, no.3, pp.694–697, Mar. 2007.

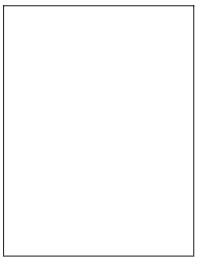
[12] Y. W. Lin, H. Y. Liu, and C. Y. Lee, "A 1-GS/s FFT/IFFT processor for UWB applications ,"IEEE J. Solid State Circuits, vol.40, no.8, pp.1726-1735, 2005.

[13] O. Gustafsson and F. Qureshi, "Addition aware quantization for low complexity and high precision constant multiplication," IEEE Signal Processing Letters, vol.17, no.2, pp.173-176, Feb. 2010.

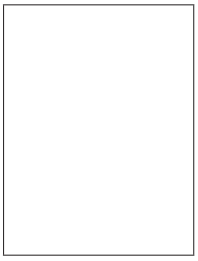
[14] O. Gustafsson, A. G. Dempster, K. Johansson, M. D. Macleod, and L. Wanhammar, "Simplified design of constant coefficient multipliers," Circuits, Systems and Signal Processing, vol.25, no.2, pp.225–251, Apr. 2006.

[15] J. Thong and N. Nicolici, "Time-efficient single constant multiplication based on overlapping digit patterns," IEEE Trans. VLSI Syst., vol.17, no.9, pp.1353–1357, Sept. 2009.

[16] O. Gustafsson, "A difference based adder graph heuristic for multiple constant multiplication problems," Proc. IEEE Int. Symp. Circuits Systems, May 2007, pp.1097–1100.



Fahad Qureshi was born in 1978. He received the M.Sc., in 2002 from NED University of Engineering and Technology Karachi, Pakistan He is currently a Ph.D. student at the Division of Electronics Systems at Linköping University, Sweden. Qureshi's research interest is design and implementation of high performance resource flexible FFTs.



Oscar Gustafsson was born in 1973. He received the M.Sc., Tekn. Lic., Ph.D., and Docent degrees in 1998, 2000, 2003, and 2008, respectively, all from Linköping University, Sweden. He is currently an Associate Professor and Head of the Division of Electronics Systems at the same university. Dr. Gustafsson's research interests is the efficient design and implementation of DSP and communication algorithms, as well as circuits and algorithms for digital arithmetic.