ELectronics EXpress

LETTER

IEICE Electronics Express, Vol.15, No.14, 1–10

Variable-length and high-precision FFT processors based on configurable constant factor multipliers and memory reallocations

Yu Xie, Xin Wei, Liang Chen, Yi-Zhuang Xie[a)], and He Chen

Beijing Key Laboratory of Embedded Real-time Information Processing Technology, Beijing Institute of Technology, Beijing 100081, China

a) *xyz551_bit@bit.edu.cn*

Abstract: A variable-length, high-precision fixed-point pipeline FFT processor design methodology is proposed in this article. As an example for synthetic aperture radar (SAR) imaging processing, a radix-2^5 single-path delay feedback (SDF) 32768-point FFT is implemented. By analyzing both the two's complement and canonic signed digit (CSD) representations of the constant factors, the proposed configurable constant factor multipliers (CCFM) can be configured to generate any constant factors applied in the radix-2^5 algorithm. The variable length architecture can be built up by a simple permutation and combination of radix-2 butterfly operations and CCFM. With the look-up table (LUT) division technique, the twiddle factor storage requirement is significantly reduced. The high precision fixed-point calculation performance is achieved based on a memory reallocation (MR) technique. When performing the non-maximum size FFT, by reallocating the idle memory resources, the fixed-point calculation precision is improved. Compared with conventional design methodology, the proposed fixed-point FFT achieves an SQNR improvement of at least 18 dB and the circuit area is reduced by at least 10%.

Keywords: FFT, variable-length, CSD, memory reallocations, SQNR

Classification: Integrated circuits

References

[1] Y.-W. Lin, *et al.*: "A 1-GS/s FFT/IFFT processor for UWB applications," IEEE J. Solid-State Circuits **40** (2005) 1726 (DOI: 10.1109/JSSC.2005.852007).

[2] B. S. Reddy and B. N. Chatterji: "An FFT-based technique for translation, rotation and scale-invariant image registration," IEEE Trans. Image Process. **5** (1996) 1266 (DOI: 10.1109/83.506761).

[3] K. Natroshvili, *et al.*: "Focusing of general bistatic SAR configuration data with 2-D inverse scaled FFT," IEEE Trans. Geosci. Remote Sens. **44** (2006) 2718 (DOI: 10.1109/TGRS.2006.872725).

[4] S. He and M. Torkelson: "Designing pipeline FFT processor for OFDM

DOI: 10.1587/elex.15.20180610
Received June 13, 2018
Accepted June 22, 2018
Publicized July 6, 2018
Copyedited July 25, 2018

(de)modulation," URSI Int. Symp. Signals, Systems, Electronics (1998) 257 (DOI: 10.1109/ISSSE.1998.738077).
[5] A. Cortes, *et al.*: "Radix-r^k FFTs: Matricial representation and SDC/SDF pipeline implementation," IEEE Trans. Signal Process. **57** (2009) 2824 (DOI: 10.1109/TSP.2009.2016276).
[6] M. Ayinala and K. K. Parhi: "FFT architectures for real-valued signals based on radix-2^3 and radix-2^4 algorithms," IEEE Trans. Circuits Syst. I, Reg. Papers **60** (2013) 2422 (DOI: 10.1109/TCSI.2013.2246251).
[7] K. J. Yang, *et al.*: "MDC FFT IFFT processor with variable length for MIMO OFDM systems," IEEE Trans. Very Large Scale Integr. (VLSI) Syst. **21** (2013) 720 (DOI: 10.1109/TVLSI.2012.2194315).
[8] C. Yang, *et al.*: "New quantization error assessment methodology for fixed-point pipeline FFT processor design," System-on-Chip Conference (SOCC) (2014) 299 (DOI: 10.1109/SOCC.2014.6948944).
[9] Y. Xie, *et al.*: "Finite word length optimization for spaceborne SAR imaging systems," IEEE Radar Conference (RadarCon) (2015) 852 (DOI: 10.1109/RADAR.2015.7131114).
[10] H. J. Kang, *et al.*: "Low complexity twiddle factor multiplication with ROM partitioning in FFT processor," Electron. Lett. **49** (2013) 589 (DOI: 10.1049/el.2013.0689).

1 Introduction

Fast Fourier transform (FFT) processors have been widely used in various applications such as communications [1], image processing [2], radar signal processing [3], etc. Pipeline architectures have an inherent advantage over other efficient hardware structures [4]. Single-path delay feedback (SDF), multi-path delay feedback (MDF) and multi-path delay commutator (MDC) architectures are usually adopted. Radix-2^k algorithm is suitable for pipeline architecture. As the radix becomes higher, the number of occupied complex multipliers decreases [5]. Instead, more constant factors appear. It is also a bottleneck for the application of higher radix FFT processors. The algorithms commonly used are radix-2^2, radix-2^3 and radix-2^4 [6].

On the other hand, fixed-point processing is faster and less resource-intensive compared with floating-point processing. Due to the finite word length effect, the optimum bit sizing technique is adopted. In [7], the input word length is fine tuned to 8-bit and the output word length is 12-bit. As for the internal word lengths, all the computations are rounded to 10-bit. Our previous work [8] has also discussed issues about the fixed-point word length configuration and signal-to-quantization-noise-ratio (SQNR) assessment.

This paper proposes a configurable constant factor multiplier to improve the flexibility of variable-length radix-2^k pipeline FFT processor. For the twiddle factor generation issue, we adopt a look-up table (LUT) division technique to reduce the hardware cost. With the proposed memory reallocation technique, the problem of increasing the feedback storage caused by a larger processing word length is solved. High precision fixed-point arithmetic performance is achieved.

DOI: 10.1587/elex.15.20180610
Received June 13, 2018
Accepted June 22, 2018
Publicized July 6, 2018
Copyedited July 25, 2018

2 Background

2.1 Requirement of SAR imaging

Large processing size, high precision and flexible processing length are three specific characteristics for FFT processors used in space-borne SAR imaging system. Different FFT sizes are shown in Table I, which depend on the different SAR imaging modes and different radar parameters. PRF stands for pulse repetition frequency of the transmitted signal. Nr and Na represent the sample numbers in range and azimuth direction respectively. Specifically, Nr_zp and Na_zp are the zero-padded points of integral power of 2, which related to Nr and Na. 2 K∼32 K-point FFTs are required for SAR imaging. Our previous work [9] analyses the necessity of adopting fixed-point processing with proper word length. 24-bit processing word length is employed in SAR imaging systems generally. Based on these requirements, we design a 2 K∼32 K variable-length, high-precision fixed-point FFT processor.

Table I. FFT sizes of some typical SAR imaging modes with different parameters

Imaging mode	Resolution (m)	*PRF*	*Nr*	*Na*	*Nr_zp*	*Na_zp*
TOPS	15	4500	2875	1600	4096	2048
Scan	5	1316	31245	5700	32768	8192
Stripmap	3	9000	5511	10000	8192	16384
Spotlight	1	4500	14846	29900	16384	32768

2.2 Review of radix-2k FFT algorithm

In order to comprehend the constant factors in radix-2^k algorithm, we review the general expression of the radix-2^p (to prevent variable confusion, we use p instead of k) algorithm. Decompose the time domain index n and frequency domain index k as follows:

$$\begin{cases} n = \dfrac{N}{2}n_0 + \ldots + \dfrac{N}{2^{p-1}}n_{p-1} + n_p;\ n_0,\ldots,n_{p-1} = 0,1\ n_p = 0,1,\ldots,\dfrac{N}{2^p} \\ k = k_0 + \ldots + 2^{p-1}k_{p-1} + 2^p k_p;\ k_0,\ldots,k_{p-1} = 0,1\ k_p = 0,1,\ldots,\dfrac{N}{2^p} \end{cases}. \quad (1)$$

By substituting the n and k expressed in (1) into W_N^{kn}, we get the expression of radix-2^p butterfly unit as follows:

$$\begin{aligned} W_N^{kn} = & \underbrace{(-1)^{n_0k_0}(-j)^{n_1k_0}}_{stage\ 1} \cdot \underbrace{(-1)^{n_1k_1}W_8^{n_2k_0}(-j)^{n_2k_1}}_{stage\ 2} \\ & \cdot \underbrace{(-1)^{n_2k_2}W_{16}^{n_3k_0}W_8^{n_3k_1}(-j)^{n_3k_2}}_{stage\ 3} \cdots \underbrace{(-1)^{n_{p-2}k_{p-2}}W_{2^p}^{n_{p-1}k_0}\cdots W_8^{n_{p-1}k_{p-3}}(-j)^{n_{p-1}k_{p-2}}}_{stage\ p-1} \quad (2) \\ & \cdot \underbrace{(-1)^{n_{p-1}k_{p-1}}}_{stage\ p} \cdot \underbrace{W_N^{n_p(k_0+2k_1+\cdots+2^{p-1}k_{p-1})}}_{\text{Twiddle factors}} \cdot W_{N/2^p}^{n_pk_p}. \end{aligned}$$

The $(-1)^{n_{p-1}k_{p-1}}$ in stage p indicates the simple radix-2 butterfly unit. The twiddle factor expression shows that the twiddle factor generation address $k_0 + 2k_1 + \cdots + 2^{p-1}k_{p-1}$ is a simple bit-reverse pattern. For a 32768-point FFT, we

DOI: 10.1587/elex.15.20180610
Received June 13, 2018
Accepted June 22, 2018
Publicized July 6, 2018
Copyedited July 25, 2018

adopt the radix-2^5 algorithm. Only three cascading radix-2^5 processing elements (PE) and two twiddle factor multiplications are required. The constant factors include: $W_8^{n_2k_0}$, $W_{16}^{n_3k_0}W_8^{n_3k_1}$ and $W_{32}^{n_4k_0}W_{16}^{n_4k_1}W_8^{n_4k_2}$.

3 Proposed architecture

3.1 Configurable constant factor multiplier (CCFM)

In order to design a CCFM, we summarize the following three rules:

a) Simplify the number of constant factors according to the trigonometric function relationship;

b) Reduce the number of adders to less than or equal to one half of the processing word length by combining both CSD and two's complement representations of constant factors;

c) Reuse the shift registers.

Table II presents the value of the constant factors in radix-2^5 algorithm. We can observe that only four cosine values and three sinusoidal values are required to express the seven complex constant factors. Constant factors (i)~(iv) are directly generated with the cosine and sinusoidal values. The other factors are generated by swapping the real and imaginary parts.

Table II. Real part and imaginary part of the complex constant factors in radix-2^5 algorithm

	Complex factors	Real part	Imaginary part
(i)	W_8^1	$\cos(\pi/4)$	$\sin(\pi/4) = \cos(\pi/4)$
(ii)	$W_8^1W_{16}^1$	$\cos(3\pi/8) = \sin(\pi/8)$	$\sin(3\pi/8) = \cos(\pi/8)$
(iii)	$W_8^1W_{32}^1$	$\cos(5\pi/16) = \sin(3\pi/16)$	$\sin(5\pi/16) = \cos(3\pi/16)$
(iv)	$W_8^1W_{16}^1W_{32}^1$	$\cos(7\pi/16) = \sin(\pi/16)$	$\sin(7\pi/16) = \cos(\pi/16)$
(v)	W_{16}^1	$\cos(\pi/8)$	$\sin(\pi/8)$
(vi)	$W_{16}^1W_{32}^1$	$\cos(3\pi/16)$	$\sin(3\pi/16)$
(vii)	W_{32}^1	$\cos(\pi/16)$	$\sin(\pi/16)$

Table III shows the 16-bit CSD and two's complement representations of the essential trigonometric function values. We adopt the CSD representation for the cosine values to decrease the number of adders occupied. We prefer a combination of CSD and two's complement representations for the sinusoidal values to reuse the shift register as much as possible. The code $\bar{1}$ stands for -1. According to the representations discussed above, the architecture of CCFM is presented in Fig. 1.

Fig. 1a is the cosine value generation block which occupies 5 adders/subtractors, 6 multiplexers and 8 shift registers. Fig. 1b is the sinusoidal value generation block which occupies 4 adders/subtractors, 4 multiplexers and 7 shift registers. With corresponding control signals, it can be configured to generate any constant factor required for the radix-2^5 algorithm. In addition, we reserve the direct paths in both Fig. 1a and Fig. 1b for the requirement of variable-length FFT architecture.

Table IV shows the circuit area of different CSD multipliers and CCFM. The areas of multipliers for implementing factors (v)~(vii) are same to those for factors (ii)~(iv) respectively. One radix-2^5 PE has five BF units, as shown in Fig. 2. From

DOI: 10.1587/elex.15.20180610
Received June 13, 2018
Accepted June 22, 2018
Publicized July 6, 2018
Copyedited July 25, 2018

Table III. 16-bit CSD/two's complement representations of the cosine and sinusoidal values

Cosine	CSD representation	Sinusoidal	CSD/two's complement
$\cos(\pi/4)$	$1.0\bar{1}0,\bar{1}010,1000,0010$	-	-
$\cos(\pi/8)$	$1.000,\bar{1}0\bar{1}0,0100,0010$	$\sin(\pi/8)$	$0.10\bar{1},0001,0000,1100$
$\cos(\pi/16)$	$1.000,00\bar{1}0,\bar{1}000,1010$	$\sin(\pi/16)$	$0.001,1001,0000,1000$
$\cos(3\pi/16)$	$1.0\bar{1}0,1010,100\bar{1},0000$	$\sin(3\pi/16)$	$0.100,100\bar{1},0010,0\bar{1}00$

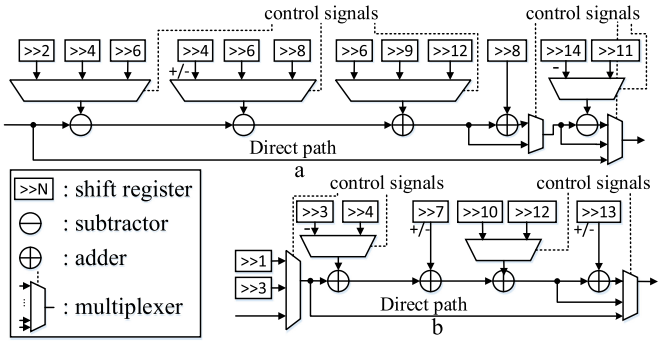

Fig. 1. Architecture of the proposed 16-bit CCFM
1a Cosine value generation block
1b Sinusoidal value generation block

Table IV. Circuit area of different CSD multipliers and CCFM

CSD constant multiplier	W_8^1	$W_8^1W_{16}^1$	$W_8^1W_{32}^1$	$W_8^1W_{16}^1W_{32}^1$	CCFM
Area (μm^2)	2966.4	3180.9	4427.5	2924.6	7316.2

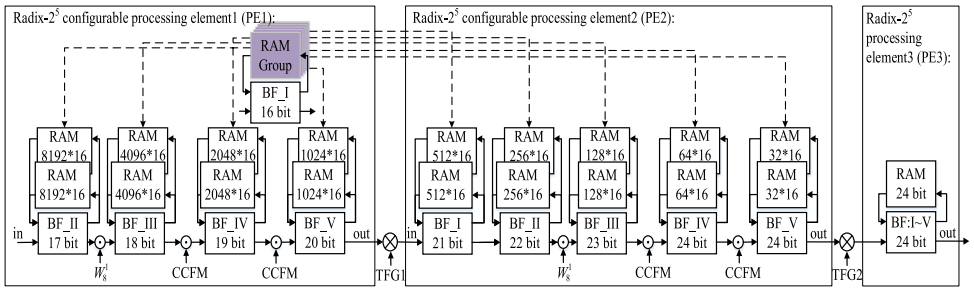

Fig. 2. Architecture of the proposed 32768-point FFT processor based on CCFM and memory reallocation (MR), working mode: 16384-point FFT

(2) we note that there is no multiplication after BF_I and only constant factor (i) multiplication is needed after BF_II. Applying CCFM to these two BF units is not area efficient. However, the multiplication after BF_III needs factors (i)~(iii) (9328.2 µm2), 22% of the circuit area is saved by applying CCFM (7316.2 µm2). The multiplication after BF_IV needs factors (i)~(vii) (24032.4 µm2), 70% of the circuit area is saved by CCFM.

3.2 Look-up table (LUT) division technique

For the twiddle factor generation after PE1 which is shown in Fig. 2 as TFG1, the depth of the ROM is 32768 when adopting an LUT-based twiddle factor generation

DOI: 10.1587/elex.15.20180610
Received June 13, 2018
Accepted June 22, 2018
Publicized July 6, 2018
Copyedited July 25, 2018

scheme. In many works, leveraging the symmetry of twiddle factors, only one quarter as much ROM space is occupied. Therefore, an 8192-depth ROM with necessary control logic is a commonly used optimized scheme. We adopt a LUT division technique [10] according to the equation as follows:

$$W^{n}_{32768} = W^{256a+b}_{32768} = W^{a}_{256} \times W^{b}_{256*128} \quad (a = 0, 1, \ldots, 63,\ b = 0, 1, \ldots, 127). \quad (3)$$

Obviously, the 32768-depth ROM is divided into two small ROMs with an additional processing overhead of one complex multiplier. Further, the symmetry of the twiddle factors is still applicable to W^{a}_{256}. Thus, the 256-depth ROM can also be replaced by a 64-depth ROM with corresponding address control logic. As a result, the demand for storage is reduced to a 192 (= 64 + 128)-depth LUT. It shows great advantages over the optimized scheme discussed above which occupies 8192-depth ROM.

For the twiddle factor generation after PE2 which is shown in Fig. 2 as TFG2, a 1024-depth ROM can be substituted by two 32-depth ROMs and necessary control logic, as shown in (4). Applying the symmetry of the twiddle factors is not cost-efficient for W^{a}_{32} due to depth of LUT.

$$W^{n}_{1024} = W^{2^5 a+b}_{1024} = W^{a}_{32} \times W^{b}_{1024} \quad (a, b = 0, 1, \ldots, 31). \quad (4)$$

On the other side, the LUT division technique is also suitable for the variable-length FFT architecture. Taking the 16384-point case as an example, (5) shows the simple principle of generating the twiddle factors with the existing LUTs. With almost no additional control logic, the compatibility of the variable-length architecture is achieved.

$$W^{n}_{16384} = W^{a}_{256} \times W^{b}_{16384} = W^{a}_{256} \times W^{2b}_{32768} \quad (a = 0, 1, \ldots, 63;\ b = 0, 1, \ldots, 63). \quad (5)$$

Table V presents the result of the three twiddle factor generation schemes mentioned above. Adopting LUT division technique in TFG1, we need two 24-bit ROMs, one is 64-depth and the other is 128-depth. The 16-bit twiddle factors are generated by these two 12-bit ROMs and multiplications. The error caused by this

Table V. Result comparison of different twiddle factor generation schemes

	LUT	Area (μm^2)	LUT+ symmetry	Area (μm^2)	LUT division	Area (μm^2)
TFG1	32768 × 32-bit ROM	224037.3	8192 × 32-bit ROM, control logic	62429.4 (0.28)	64 × 24-bit ROM, 128 × 24-bit ROM, 12-bit real multiplier×3, 16-bit real adder×5, control logic	20209.4 (0.09)
TFG2	1024 × 32-bit ROM	19156.3	256 × 32-bit ROM, control logic	12770.8 (0.67)	32 × 24-bit ROM×2, 12-bit real multiplier×3, 16-bit real adder×5, control logic	7654.4 (0.40)

DOI: 10.1587/elex.15.20180610
Received June 13, 2018
Accepted June 22, 2018
Publicized July 6, 2018
Copyedited July 25, 2018

way is about 10^{-4} compared with direct LUT approach, which is acceptable. A complex multiplier implemented by three 12-bit real multipliers and five 16-bit real adders is also adopted, which can further save circuit area compared with the traditional four real multipliers scheme.

Process of twiddle factors generation in TFG2 is similar to that in TFG1. The circuit area can be significantly decreased by applying LUT division technique. Specifically, 91% of the area is saved in TFG1 and 60% in TFG2 compared with direct LUT approach. Even compared with the one quarter symmetric LUT approach, 67% of the area is saved in TFG1 and 40% in TFG2.

3.3 Memory reallocation (MR) technique

A huge challenge for fixed-point FFT processor design is the problem of finite word-length effect. According to the basic rules of fixed-point operations, if the carry bit is reserved after an addition, no quantization noise will be introduced into the fixed-point processing system. However, longer processing word length means more memory occupations for a pipeline FFT processor. Taking a 1024-point FFT with 12-bit input as an example, there are two common word-length configuration modes. Mode 1 customizes 12-bit during the whole processing. The SQNR decreases due to the quantization noise. Mode 2 expands 1 bit after every BF unit (input 12-bit and output 21-bit). About 8.1% of extra memories will be occupied. The SQNR remains much higher. However, as the FFT size becomes larger, the increase in memory will be untenable.

We adopt an MR technique to solve the above issue. When processing the non-maximum size FFT, the first BF unit (BF_I) in PE1 and the corresponding feedback RAM are in idle state. Thus we divide this 16384-depth RAM into a RAM group. The RAM group consists of ten smaller RAMs. 8192-depth, 4096-depth, 2048-depth and 1024-depth RAMs are reallocated to the first PE (PE1), 512-depth, 256-depth, 128-depth, 64-depth and one of two 32-depth RAMs are reallocated to the second PE (PE2). Still taking 16384-point FFT as an example, the four RAMs reallocated to PE1 and the five RAMs reallocated to PE2 will be assigned to the four BF units in PE1 and the five BF units in PE2 respectively. Thus, the feedback RAMs of the BF units are enlarged. Through the bit-width extension of the feedback RAMs, the calculation results of larger word-length can be stored. The reservation of larger processing word lengths can improve the precision of the whole FFT processing. Fig. 2 shows the memory reallocating process.

The word lengths of the BF units in PE1 and PE2 are customized as shown in Fig. 2, which increase 1 bit after every BF unit (from 16-bit to 24-bit). Word lengths of the BF units in PE3 are all customized 24-bit. Owing to the very small depth of RAMs in PE3 (maximum depth is 16), 24-bit processing word length in PE3 causes little resource occupation increase compared with 16-bit.

Therefore, when performing 32768-point FFT, the input word length is 16 bit. The internal processing word lengths of PE1 and PE2 are all 16-bit, limited by the capacity of RAMs. The output word length is 24 bit. When performing non-32768-point FFT, the output of PE2 is expanded to 24-bit based on the memory reallocation technique. The word-length schemes of different FFT sizes are detailed in Table VI.

DOI: 10.1587/elex.15.20180610
Received June 13, 2018
Accepted June 22, 2018
Publicized July 6, 2018
Copyedited July 25, 2018

Table VI. Word-length schemes under different FFT sizes with the memory reallocation technique

FFT Size	Word-length Scheme										
	PE1 and PE2										PE3
32768	16	16	16	16	16	16	16	16	16	16	24
16384		17	18	19	20	21	22	23	24	24	24
8192			18	19	20	21	22	23	24	24	24
4096				19	20	21	22	23	24	24	24
2048					20	21	22	23	24	24	24

We compare the circuit area of real adders and memories applying MR technique with non-MR technique. The result is shown in Table VII. The extra circuit area of adders is about 35% and that of memories is about 1.6%. The total extra overhead of MR scheme is no more than 1.75%. The significant increase of calculation accuracy resulted from MR scheme will be presented in section IV. Although the adders with a larger word length may occupy more circuit area, taking into account that the circuit area of an adder is not very sensitive to the word length and that the reallocation scheme results in processing precision improvement, this trade-off is cost-effective. Therefore, MR technique achieves high-precision (measured by SQNRs) with lower overhead of circuit area.

Table VII. Area comparison of the real adders and memories under different schemes

Scheme	Real adders		Area (μm^2)	Memories (RAMs)*			Area (μm^2)
	Bit width	Amount		Bit width	Depth	Amount	
No MR	16-bit	30	9518.4	32	16384	1	2985222.4
					8192	1	
					4096	1	
					1024	1	
					2048	1	
					512	1	
					256	1	
					128	1	
With MR	16-bit	2	12894.4	32	8192	2	3033411.2
	17-bit	2			4096	2	
	18-bit	2			2048	2	
	19-bit	2			1024	2	
	20-bit	2			512	2	
	21-bit	2			256	2	
	22-bit	2			128	2	
	23-bit	2			control logic	/	
	24-bit	14					

*The RAMs with the depth less than 128 are implemented with distributed logic.

DOI: 10.1587/elex.15.20180610
Received June 13, 2018
Accepted June 22, 2018
Publicized July 6, 2018
Copyedited July 25, 2018

3.4 Configurable processing elements (PE)

To achieve the variable-length architecture of FFT (from 2 K to 32 K), six multiplexers are inserted to the radix-2^5 processing element as shown in Fig. 3. The maximum 32 K-point FFT is implemented without any bypass. When the FFT size varies, the number of BF units occupied and the distribution of constant twiddle factors should change accordingly. The data stream is controlled by the six multiplexers. The constant factors corresponding to different FFT size are generated by the CCFM. Thus, 2 K, 4 K, 8 K and 16 K-point FFT are realized.

Finally, the proposed 32768-point variable-length FFT processor is implemented by cascading configurable processing elements (PE1, PE2 and PE3) and LUT-division-based twiddle factor generation blocks discussed above (TFG1 and TFG2).

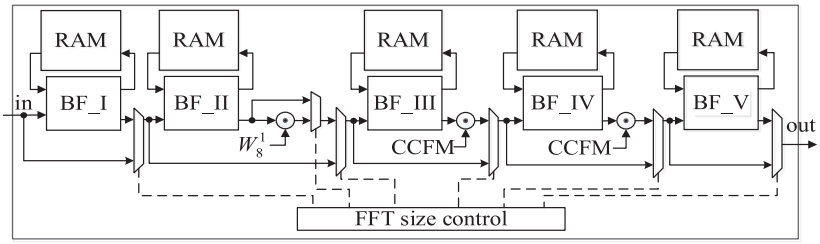

Fig. 3. Configurable processing element (PE) of the variable-length FFT.

4 Result and comparison

Limited by the lack of literatures on large-point FFT implementation, we replicate several conventional radix-2^k SDF FFT processors under 65 nm CMOS technology. For fairly comparison, the input word lengths is set 16 bit and the output word length is set 24 bit. The internal word length scheme is same as the 32768-case shown in Table VI. The word length of the twiddle factors is also set 16-bit. The one quarter LUT-based method is adopted for the twiddle factor generation.

Table VIII shows pre-layout area comparison result. The circuit area is reduced by at least 13%. The SQNR performance in the case of 32768-point is better than others because there are less twiddle factor multiplications in the radix-2^5 algorithm. When performing non-32768-point FFT the proposed fixed-point FFT achieves an SQNR improvement of at least 18 dB.

Table VIII. Result comparison with several conventional 16-bit 32768-point variable-length fixed-point pipeline FFT processors.

	Area (mm^2)	SQNR (dB) (test input: white Gauss noise)				
		32768	16384	8192	4096	2048
R2SDF	20.6	46.5	49.9	53.1	54.4	58.1
R2^2SDF	17.5	49.7	53.2	56.3	58.4	62.8
R2^3SDF	16.1	50.8	54.2	57.2	59.5	64.3
Proposed	14.0	51.5	79.2	80.5	81.9	82.4

DOI: 10.1587/elex.15.20180610
Received June 13, 2018
Accepted June 22, 2018
Publicized July 6, 2018
Copyedited July 25, 2018

5 Conclusion

This paper proposes a 32768-point variable-length pipeline FFT processor. With the CCFM, LUT division and memory reallocation techniques, the circuit area is reduced and the fixed-point SQNR performance is improved. The design methodology can be expanded to higher radix-2^k processor implementations.

Acknowledgements

This study is supported by the National Natural Science Foundation of China under Grant 91438203, the National Key R & D Program of China under contract #2017YFB0502800 and #2017YFB0502804.

DOI: 10.1587/elex.15.20180610
Received June 13, 2018
Accepted June 22, 2018
Publicized July 6, 2018
Copyedited July 25, 2018