SCIENCE CHINA
Information Sciences

• LETTER •

February 2018, Vol. 61 029301:1–029301:3
doi: 10.1007/s11432-017-9132-x

A novel word length optimization method for radix-2^k fixed-point FFT

Chen YANG, Yizhuang XIE & He CHEN*

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

Received 18 April 2017/Accepted 18 May 2017/Published online 13 September 2017

Citation Yang C, Xie Y Z, Chen H. A novel word length optimization method for radix-2^k fixed-point FFT. Sci China Inf Sci, 2018, 61(2): 029301, doi: 10.1007/s11432-017-9132-x

Dear editor,

Fast Fourier transform (FFT) is one of the most fundamental algorithms used in digital signal processing. Many applications such as orthogonal frequency division multiplexing (OFDM), long term evolution (LTE), and ultra-wideband (UWB) systems require an area efficient, high accuracy FFT processor. To design a high-precision and low-complexity FFT/IFFT processor architecture, the optimum bit sizing technique in each stage is usually adopted. Many fixed-point pipeline FFT processors are designed in previous studies [1–5]. However, most of the word length schemes in these studies are proposed based on long-time fixed-point simulation. It is difficult to provide an accurate, fast word length scheme because of the diversity of FFT algorithms and the complexity of circuit structure. In this letter, we focus on the widely-used radix-2^k decimation-in-frequency (DIF) fast Fourier transform (FFT) algorithm. Based on our previous research on fixed-point FFT signal-to-quantization-noise ratio (SQNR) assessment [6], the analytical expression of the word length in different stages is deduced. We further put forward a word length optimization method based on the analytical expression.

In our previous work [6], we reached an SQNR analytical expression of radix-2^2 fixed-point FFT. We re-list the output SQNR expression as follows:

$$\mathrm{SQNR}=\frac{P_X}{P_E}=\frac{P_X}{P_A+P_M}=\frac{N\cdot(\frac{1}{4})^{\sum_{i=1}^{v}T_i}\sigma_x^2}{N\cdot\sum_{i=1}^{v}(\frac{1}{4})^{\sum_{j=i+1}^{v}T_j}2^{v-i}\sigma_{ai}^2+\sum_{i=1}^{v}(\frac{1}{4})^{\sum_{j=i+1}^{v}T_j}2^{v-i}\sigma_{mi}^2}. \tag{1}$$

The variables are defined as follows:

- σ_x^2 is the variance of input signal.
- σ_{ai}^2 is the addition noise variance in stage i.
- σ_{mi}^2 is the complex multiplication noise variance in stage i.
- b_0 is the initial input word length of FFT and b_i is the word length in stage i ($i=1,2,\ldots,v; v=\log_2 N$).
- T_i is the word length scaling variable in stage i.

According to addition operation rules, word length is expected to increase by 1 bit after one addition. Thus we define $T_i=0$ if the word length increases by 1 bit after the butterfly operation in stage i. The relationship between b_0, b_i and T_i is described as follows:

$$b_i=b_0+i-\sum_{j=1}^{i}T_j. \tag{2}$$

In order to establish the relationship between

* Corresponding author (email: chenhe@bit.edu.cn)
The authors declare that they have no conflict of interest.

quantization noise variance and word length, we analyze the rounding and truncation issues based on the assumptions proposed in [7]. The addition noise variance in both rounding and truncation issues is expressed as follows:

$$\sigma_{ai}^2 = \begin{cases} N \cdot \alpha_i \cdot 2^{-2b_i}/12, & \text{rounding}, \\ N \cdot \alpha_i \cdot 2^{-2b_i}/3, & \text{truncation}. \end{cases} \tag{3}$$

The variable α_i is defined according to the addition operation rules as follows:

$$\alpha_i = \begin{cases} 1, & b_i < b_{i-1} + 1, \\ 0, & b_i = b_{i-1} + 1. \end{cases} \tag{4}$$

Complex multiplication is usually composed of four real multiplications. In addition, we usually ensure that the data word length remains unchanged after a multiplication operation. Thus, the multiplication noise variance in both rounding and truncation issues can be expressed as follows:

$$\sigma_{mi}^2 = \begin{cases} n_i \cdot 2^{-2b_i}/3, & \text{rounding}, \\ n_i \cdot 4 \cdot 2^{-2b_i}/3, & \text{truncation}. \end{cases} \tag{5}$$

n_i is the number of non-trivial twiddle factors in the radix-2^k algorithm. We revealed the value of n_i in Appendix A.

Although (1) is extended to both rounding and truncation issues, it is still not complete. The total quantization noise should consist of two parts. One part is the quantization noise generated by the internal arithmetic operations of fixed-point FFT. Another is the initial inherent quantization noise associated with the input fixed-point data. The quantization noise power of the input b_0-bit fixed-point data can be expressed as follows:

$$P_{E\_ini} = \begin{cases} 2^{-2b_0}/12, & \text{rounding}, \\ 2^{-2b_0}/3, & \text{truncation}. \end{cases} \tag{6}$$

Thus, the quantization noise power P_E should be the sum of addition noise power P_A, multiplication noise power P_M, and the initial quantization noise power P_{E_ini}. By substituting (3), (5) and (6) into (1), the modified SQNR assessment expression is described as follows:

$$\text{SQNR} = \begin{cases} \dfrac{(1/4)^{\sum_{i=1}^{v} T_i}}{1 + \sum_{i=1}^{v} (1/4)^{\sum_{j=i+1}^{v} T_j - \sum_{k=1}^{i} T_k} \cdot N\alpha_i \cdot 2^{-3i} + \sum_{i=1}^{v} (1/4)^{\sum_{j=i+1}^{v} T_j - \sum_{k=1}^{i} T_k} \cdot 4n_i \cdot 2^{-3i}} \cdot \dfrac{\sigma_x^2}{2^{-2b_0}/12}, & \text{rounding}, \\ \dfrac{(1/4)^{\sum_{i=1}^{v} T_i}}{1 + \sum_{i=1}^{v} (1/4)^{\sum_{j=i+1}^{v} T_j - \sum_{k=1}^{i} T_k} \cdot N\alpha_i \cdot 2^{-3i} + \sum_{i=1}^{v} (1/4)^{\sum_{j=i+1}^{v} T_j - \sum_{k=1}^{i} T_k} \cdot 4n_i \cdot 2^{-3i}} \cdot \dfrac{\sigma_x^2}{2^{-2b_0}/3}, & \text{truncation}. \end{cases} \tag{7}$$

Different radix-2^k algorithms correspond to the different values of n_i. Thus, the modified SQNR analytical form (7) is suitable for radix-2^k algorithms.

Define that

$$P = \sum_{i=1}^{v-1} (1/4)^{\sum_{j=i+1}^{v-1} T_j - \sum_{k=1}^{i} T_k} \cdot \alpha_i \cdot 2^{-3i}, \tag{8}$$

$$Q = (1/4)^{-\sum_{i=1}^{v-1} T_i}, \tag{9}$$

$$\text{SQNR}_0 = \begin{cases} 12 \cdot \sigma_x^2/2^{-2b_0}, & \text{rounding}, \\ 3 \cdot \sigma_x^2/2^{-2b_0}, & \text{truncation}, \end{cases} \tag{10}$$

$$R = \text{SQNR}_0/\text{SQNR}. \tag{11}$$

The expression of the word length scaling variable T_i is derived as follows. The detailed derivation process is described in Appendix B.

$$T_i = \begin{cases} \dfrac{1}{2}\log_2\left(\dfrac{R}{Q} - P\right), & \alpha_i = 0, \\ \dfrac{1}{2}\log_2\left(\dfrac{-1 + \sqrt{1 - 4\alpha_i 2^{-3i} \cdot (Q \cdot P - R)}}{2\alpha_i 2^{-3i} \cdot Q}\right), & \alpha_i \neq 0. \end{cases} \tag{12}$$

The current stage scaling variable T_i is closely related with b_0, SQNR, and the scaling variables of previous stages: $\{T_1, T_2, \ldots, T_{i-1}\}$. By substituting (12) into (2), the presentation of internal word length sequence $\{b_i\}$ is finally obtained.

According to the derivation above, the internal word length $\{b_i\}$ can be directly calculated. We set up a recursive feedback mechanism to generate the word length scheme $\{b_i\}$ according to a set of constraints. This mechanism is summarized as a word length optimization method as follows:

(1) Input constraints. b_0, SQNR, NFFT and quantization mode (rounding/truncation).

Table 1 Wordlength optimization of a 16384-point fixed-point FFT

Schemes	b_0 b_1 b_2 b_3 b_4 b_5 b_6 b_7 b_8 b_9 b_{10} b_{11} b_{12} b_{13} b_{14}	Memory (bit)
Regular	24 24 24 24 24 24 24 24 24 24 24 24 24 24 24	786432
Proposed	16 17 18 18 19 20 21 21 22 23 24 25 26 27 27	581004

(2) Assign initial value. $\mathrm{SQNR}_{\mathrm{ini}} = \mathrm{SQNR}$.

(3) Calculate $\{T_i\}$ with $\mathrm{SQNR}_{\mathrm{ini}}$ using (12).

(4) Substitute $\{T_i\}$ into (7) to obtain $\mathrm{SQNR}_{\mathrm{est}}$.

(5) Calculate the SQNR error of current solution $\{T_i\}$ by $\mathrm{SQNR}_{\mathrm{err}} = \mathrm{SQNR}_{\mathrm{est}} - \mathrm{SQNR}$.

(6) Revise $\mathrm{SQNR}_{\mathrm{ini}}$ by $\mathrm{SQNR}_{\mathrm{ini}} = \mathrm{SQNR}_{\mathrm{ini}} - \mathrm{SQNR}_{\mathrm{err}}$.

(7) Repeat (3)–(6) until the SQNR error is less than a threshold.

(8) Transform $\{T_i\}$ to $\{b_i\}$ using (2).

(9) Output $\{b_i\}$.

It is worth mentioning that the whole procedure of the above described method is based on the derived equations. The convergence time of this recursive process depends on the threshold. It takes little time to complete the recursive steps and obtain the word length scheme $\{b_i\}$.

Applying the optimum bit sizing technique to large-size FFT undoubtedly saves more hardware resources. Thus, in order to validate the effectiveness of our method, we consider a 16384-point FFT word length optimization issue as an example.

Table 1 shows the comparison between the word length scheme generated by the proposed method and that usually adopted. The SQNR performance of the two schemes are both about 70.1 dB. Our method significantly reduces the memory usage by 26.1% compared with the 24-in-24-out scheme. The calculation of the memory bit count is shown in Appendix C.

Conclusion and future work. In this letter, we extend the SQNR assessment to the radix-2^k algorithm under both rounding and truncation cases. We further derive the analytical word length expression based on this modified SQNR assessment expression. A word length optimization method is proposed accordingly. Considering a 16384-point fixed-point FFT as an example, an optimized word length scheme is achieved. In conclusion, the proposed method rapidly and accurately generates word length optimization schemes that realize an efficient trade-off between FFT performance and hardware expenditure.

Acknowledgements This work was supported by Chang Jiang Scholars Programme (Grant No. T2012122), and Hundred Leading Talent Project of Beijing Science and Technology (Grant No. Z141101001514005).

Supporting information Appendixes A–C. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

1 Yu C, Yen M H. Area-efficient 128- to 2048 1536-point pipeline FFT processor for LTE and mobile WiMAX systems. IEEE Trans Very Large Scale Integr Syst, 2015, 23: 1793–1800

2 Ayinala M, Parhi K K. FFT architectures for real-valued signals based on radix-2^3 and radix-2^4 algorithms. IEEE Trans Circ Syst, 2013, 60: 2422–2430

3 Cho T, Lee H. A high-speed low-complexity modified radix-2^5 FFT processor for high rate WPAN applications. IEEE Trans Very Large Scale Integr Syst, 2013, 21: 187–197

4 Ma C M, Chen H, Yu J Y, et al. A novel conflict-free parallel memory access scheme for FFT constant geometry architectures. Sci China Inf Sci, 2013, 56: 042404

5 Ren H Y, Wang Y Q, Jiang L, et al. CW interference mitigation in GNSS receiver based on frequency-locked loop. Sci China Inf Sci, 2016, 59: 082201

6 Yang C, Xie Y, Chen H, et al. New quantization error assessment methodology for fixed-point pipeline FFT processor design. In: Proceedings of the 27th IEEE International IEEE System-on-Chip Conference (SOCC), Las Vegas, 2014. 299–305

7 Oppenheim A V, Weinstein C J. Effects of finite register length in digital filtering and the fast Fourier transform. Proc IEEE, 1972, 60: 957–976