FDOCT Imaging Processor for Portable OCT Systems with High Imaging Rate

Song-Nien Tang*1, Chih-Yu Hsiang1, Sheng-Jie Huang1 and, Wan-Wei Chen1

1 Department of Information and Computer Engineering, Chung Yuan Christian University, Chung Li District, Taoyuan City, Taiwan 32023, R.O.C

a) sntang@ice.cycu.edu.tw

Abstract: Frequency-domain optical coherence tomography (FDOCT) has been widely applied in medical image inspection. This paper presents an image formation processor capable of performing all FDOCT imaging operations, including DC noise removal, re-sampling, real-valued fast Fourier transform (RFFT), and display processing. A hardware-efficient RFFT unit and memory-based display processing engine make it possible to generate gray-scale display data at high OCT imaging rates. System-level design verification was performed using an FPGA platform and a mobile phone to evaluate the efficacy of the proposed scheme.

Keywords: Optical Coherence Tomography, OCT, FFT

Classification: Integrated circuits (memory, logic, analog, RF, sensor)

References

[1] J. M. Schmitt, Optical Coherence Tomography (OCT): a Review, IEEE J. Sel. Topics Quantum Electron., vol. 5, no. 4, pp. 1205-1215, July/Aug. 1999.

[2] P. H. Tomlins and R. K. Wang, Theory, development and applications of optical coherence tomography, J. of Physics D: Applied Physics, vol. 38, pp. 2519-2535, 2005.

[3] R. Leitgeb, C. K. Hitzenberger and A. F. Fercher, Performance of fourier domain vs. time domain optical coherence tomography, Opt. Express, vol. 11, no. 8, pp. 889894, Apr. 2003.

[4] M. Ali and R. Parlapalli, Signal processing overview of optical coherence tomography systems for medical imaging Texas Instruments, Inc., White Paper, 2010. Available: http://www.ti.com/lit/wp/sprabb9/sprabb9.pdf

[5] Y. Yasuno, V. D. Madjarova, S. Makita, M. Akiba, A. Morosawa, C. Chong, T. Sakai, K.-P. Chan M. Itoh and T. Yatagai, Three-Dimensional and High-Speed Swept-Source Optical Coherence Tomography for in vivo Investigation of Human Anterior Eye Segments, Optics Express. vol. 13, no. 26, pp. 1065210664, Dec. 2005.

[6] Kang Zhang, and Jin U. Kang, Graphics Processing Unit-Based Ultrahigh Speed Real-Time Fourier Domain Optical Coherence Tomogra-
phy, *IEEE J. Sel. Topics Quantum Electron.*, vol. 18, no. 4, pp. 1270-1279, July/Aug. 2012.

[7] Chen D. Lu, M. F. Kraus, B. Potsaid, J. J. Liu, W. Choi, V. Jayaraman, Alex E. Cable, J. Hornegger, J. S. Duker and J. G. Fujimoto, Handheld Ultrahigh Speed Swept Source Optical Coherence Tomography Instrument using a MEMS Scanning Mirror, *Biomed. Optics Express*, vol. 5, no. 1, pp. 293-311, Jan., 2014.

[8] S. Schneider, M. Lauermann, P.-I. Dietrich, C. Weimann, W. Freude and C. Koos, Optical Coherence Tomography System Mass Producible on a Silicon Photonic Chip, *Optics Express*, vol. 24, no. 2, pp. 1573-1586, Jan., 2016.

[9] A. E. Desjardins, B. J. Vakoc, M. J. Suter, S.-H. Yun, G. J. Tearney and B. E. Bouma, Real-time FPGA Processing for High-Speed Optical Frequency Domain Imaging, *IEEE Trans. Medical Image*, vol. 28, no. 9, pp. 14681472, Sep. 2009.

[10] W. W. Smith and J. M. Smith, Handbook of Real-Time Fast Fourier Transforms: Wiley-IEEE, 1995.

[11] Ruikang K Wang, Fourier Domain Optical Coherence Tomography Achieves Full Range Complex Imaging in vivo by Introducing a Carrier Frequency during Scanning, *Physics in Medicine and Biology*, vol. 52, no. 19, pp. 58975907, 2007.

[12] L. An, P. Li, T. T. Shen, and R. Wang, High Speed Spectral Domain Optical Coherence Tomography for Retinal Imaging at 500,000 A-lines per Second, *Biomed. Optics Express*, vol. 2, no. 10 pp. 2770-2783, Oct. 2011.

[13] H. Shousheng and M. Torkelson, Designing pipeline FFT processor for OFDM (de)mulation, in *Proc. URSI Int. Symp. Signals, Syst. Electron.*, 1998, pp. 257262.

[14] J.-Y. Oh and M. S. Lim, Area and power efficient pipeline FFT algorithm, in *Proc. IEEE Workshop SIPS*, 2005, pp. 520525.

[15] S.-N. Tang, C.-H. Liao, and T.-Y. Chang, An area- and energy-efficient multimode FFT processor for WPAN/WLAN/WMAN systems, *IEEE J. Solid-State Circuits*, vol. 47, no. 6, pp. 14191435, Jun. 2012.

[16] N. H. E. Weste and D. Harris, *CMOS VLSI Design: A Circuits and Systems Perspective*. Boston, MA: Addison-Wesley, 2004.

[17] M. Mansour, A. M. El-Sawy, M. S. Aziz, and A. T. Sayed A New Hardware Implementation of Base 2 Logarithm for FPGA, *International Journal of Signal Processing Systems*, vol. 3, no. 2, Dec. 2015.

1 Introduction

Optical coherence tomography (OCT) [1] has been widely applied in medical image inspection to provide cross-sectional profile images [1], [2] of surface tissue, such as the eye and oral cavity at m-level resolutions with mm-level field depth. OCT technology can be classified as time-domain OCT (TDOCT) and Fourier-domain OCT (FDOCT), which is superior [2]-[4]. As shown in Fig. 1, most FDOCT systems use a swept source laser [2], [4] (i.e., SS-OCT), involving three functional blocks: an interferometer, a coherent receiver, and image formation. In the interferometer, the beam from the laser source is split and then respectively transmitted to a tissue sample and a reference mirror.
Fig. 1. Block diagram of a typical FDOCT system using the swept source laser (SS-OCT).

In the coupler, the reflected beams from the mirror and the sample are used to generate an interference signal $I(\lambda)$, which is a wavelength-domain optical signal when the wavelength of the laser source is swept (triggered by controls). The $I(\lambda)$ signal is then sensed by a photo detector, electrically amplified, and sampled by an analog-to-digital converter (ADC) in a coherent receiver. The sampled $I(\lambda)$ is sent to the image formation block to perform OCT imaging operations and generate the display data. By rotating the scanning mirror (Fig. 1), the captured $I(\lambda)$ becomes a line of axial signals (an A-line) capable of generating one-dimensional (1D A-Scan) depth-resolved structural information of the sample, following image formation. By combining several lines of A-Scan data in the lateral direction, a frame of two-dimensional (2D B-Scan) OCT image or even several frames of a 3D C-Scan OCT image can be obtained. Most conventional cart-based FDOCT systems tend to be cumbersome, due to the optical-mechanical front end (interferometer and coherent receiver) and the need for a separate PC to perform image formation [5], [6]. However, the recent trend has been toward small-scale apparatus for mobile medical applications. This has led to the development of micro-electromechanical systems (MEMS) [7] and the photonic-integrated circuit (PIC) [8] technology for the FDOCT optical-mechanical front end. Thus, an FDOCT imaging scheme based on an area- and energy-efficient processor, such as field programmable gate array (FPGA) or application-specific integrated circuit (ASIC), could be of considerable practical value in the development of small-scale handheld FDOCT devices. Moreover, there is a need for hardware-based FDOCT image formation systems capable of high imaging rates [9]. This paper proposes an FDOCT imaging processor capable of rapid image formation specifically for portable OCT applications.

2 FDOCT image formation

Figure 2 presents the operation flow of FDOCT image formation, which consists of DC removal, re-sampling, fast Fourier transform (FFT), and display processing. Interference signal $I(\lambda)$ (Fig. 1), usually contains DC background noise, which means that the DC term should be removed in the first stage.
The DC-removed interference signal, \( \Gamma'(\lambda) \), is in the wavelength domain (\( \lambda \)-space); therefore, a re-sampling operation \([3],[5]\) is performed to convert \( \Gamma'(\lambda) \) into \( \Gamma'(w) \), which is a linear signal in the frequency domain (k-space) representing the spectrum of an A-line signal. A depth-resolved profile (corresponding to time) can be obtained by applying an FFT (or the inverse, IFFT) operation to \( \Gamma'(w) \) \([1]-[4]\). The FFT results are then sent for processing to generate display data with the magnitude of FFT outputs compressed to fit an appropriate dynamic range \([4],[6]\). The interference signal is a series of real-valued data; therefore, without the application of signal processing, the FFT inputs in Fig. 2 are real numbers. This means that the corresponding FFT execution is a real-valued FFT (RFFT) operation \([2]-[4]\). When applying RFFT to a real N-point sequence, \( x(n) \), the FFT results, \( X(k) \) theoretically exhibit Hermitian symmetry for \( X(k) \) and \( X(-k) \). Thus, only half (i.e. \( N/2 \)) of FFT output samples are used in A-Scan display data.

**Fig. 2.** Block diagram showing FDOCT image formation.

### 3 Proposed FDOCT imaging processor

The idea behind computing RFFT is to use a complex-valued FFT (CFFT) in which the imaginary part of the input is set to zero. This scheme is simple; however, it does not make efficient use of CFFT hardware resources. One existing method takes hardware efficiency into account by using CFFT to calculate two RFFTs simultaneously \([10]\). For example, the two original N-point RFFTs of two real-valued sequences, \( g(n) \) and \( h(n) \) (\( n=0,1,,N-1 \)), could be alternatively computed by applying an N-point CFFT to a complex sequence, \( f(n)=g(n)+j* h(n) \). If we let \( G(k) \) and \( H(k) \) are the positive N/2 Hermitian symmetry parts of the FFT for \( g(n) \) and \( h(n) \), respectively, and \( F(k) \) is the N-point FFT output of \( f(n) \). The \( G(k) \) and \( H(k) \) results are obtained via post-processing of \( G(k)=1/2*[F(k)+F'(-k)] \) and \( H(k)=(-j/2)*[F(k)-F'(-k)] \), where \( F'(.) \) refers to the complex conjugate term. The above-mentioned strategy makes it possible to use two streams of N-point RFFT instead of a single one-stream N-point CFFT in common with post-processing, which is applicable to CFFT-based hardware-efficient RFFT computation. We developed an FDOCT image formation processor (Fig. 3), to enable parallel imaging operations on two interference signal streams. After the interface unit, there are four modules respectively used for DC removal and re-sampling as well as a CFFT-based RFFT unit, and a unit for display processing, as described below.
3.1 DC removal
DC noise measurement results (i.e., the DC term) are pre-stored in RAM and subtracted directly from the interference signal.

3.2 Re-sampler
Generally, floating-point re-sampling indices for OCT systems are first obtained via a calibration mechanism during the initial set-up [5], [11]. Interpolation is applied to two adjacent data samples indexed by the two integers closest to the re-sampling index in order to obtain re-sampled data. Figure 4 illustrates the architecture of the proposed re-sampler used for linear interpolation (including front-end DC removal). Two sets of memory modules take turns storing the DC-removed input data stream by stream, whereupon the odd- or even-indexed data in each stream is stored separately. Based on the value of the re-sampling index (pre-stored in RAM), we determine the two closest integer indices (odd or even) and their corresponding interpolated coefficients in order to access two adjacent data streams for linear interpolation operations.

3.3 CFFT-based RFFT Computation
As mentioned above, we employed an N-point CFFT in conjunction with post-processing operations to realize the two-stream N-point RFFTs (i.e., CFFT-based RFFT computation). Generally, the number of FFT points N is 512, 1024, or 2048, depending on the number of samples selected from an A-line signal [2], [12]. Figure 5 presents an example of the proposed...
CFFT-based RFFT hardware architecture using 1024-point FFT. We take throughput into account by employing a single-path delay feedback (SDF) pipelined FFT structure based on the radix-2^k algorithm, where k is a positive integer [13], [14]. For the 1024-point CFFT computation, there are respectively three, four, and three radix-2 operation stages in the SDF architecture, which correspond to the radix-2^3/2^4/2^3 FFT algorithm, as shown in Fig. 5. Each radix-2 stage includes a BF2/IO unit, which performs addition and subtraction of the input for radix-2 execution (BF2 mode) or routes the input and the FIFO data for In/Out delivery (IO mode). The radix-2^k-based pipelined FFT architecture helps to reduce the costs associated with multiplication hardware [13]-[15]. In each set of radix-2^k stages (k = 3 or 4), only an area-efficient constant multiplier is required, and complex multipliers are employed only among the radix-2^k-stage sets. When using the radix-2^k FFT algorithm, the SDF CFFT block sequentially sends the output, F(k), with k in a bit-reversed order [14]. In this way, two sets of 512-element FIFOs are used to alternately store each stream of CFFT results for the permutation of F(k) and F(-k) in a normal order. This makes it possible for the post-processing block to perform the associated addition or subtraction functions to sequentially generate the required G(k) and H(k).

Compared to the direct N-point RFFT approach in which zero is assigned to the imaginary part of the CFFT input, the proposed CFFT-based RFFT computation doubles the throughput by introducing two-stream operations. Furthermore, all N samples at the output are used for the A-Scan display (i.e., 100% utility), compared with the direct N-point RFFT approach in which only N/2 data samples are exploited (i.e., 50% utility). The proposed CFFT-based RFFT unit is suitable for OCT systems requiring high A-line throughput. There are two application scenarios. 1) Dual-camera FDOCT systems in which A-line signals follow dual optical paths [12]. The proposed design is able to feed two FDOCT image formation paths for two-stream operations. 2) OCT systems employing a camera with a high A-line scanning rate (when using a single optical path) [6]. Our processor can be operated at a relatively slow clock rate (approximately half the sampling frequency of the A-line signal) in order to reduce power consumption.

**Fig. 5.** Block diagram showing hardware architecture of proposed CFFT-based hardware-efficient RFFT computation unit.
3.4 Memory-based Display Processing

In our design, display processing involves the following three operations:

i) Magnitude computation: Assuming that the RFFT output value is $a + jb$, then $a^2 + b^2$ (or its square root) is calculated as a magnitude.

ii) Logarithm compression: The magnitude is normalized (i.e., $\text{nor.}(a^2 + b^2)$) to within 0 and 1, and then compressed via the $10 \cdot \log_{10}[\text{nor.}(a^2 + b^2)]$

iii) Gain and dynamic range mapping: The logarithm compression value ($LC_{\text{val}}$) is converted (mapped) into 8-bit gray scale for image display for a given gain offset and dynamic range ($DR_{dB}$).

A hardware-based design makes it possible to implement items i) and ii) using an arithmetic circuit or a memory-based lookup table to obtain magnitude and logarithm values [16], [17]. For FPGA-based designs, we employ the lookup table approach to take advantage of the large number of usable memory devices that can be embedded in an FPGA chip. To reduce table capacity, we apply a dynamic scaling scheme [15] to the RFFT output ($a + jb$), wherein data is represented in floating-point form ($2^E$, where M is the data mantissa and E is the data exponent), as shown in Eq. (1). Thus, operations i) and ii) can be performed using Eq. (2) to obtain the $LC_{\text{val}}$. Based on Eq. (2), we can use ROM as a lookup table for the squared value (Eq. (2)-A and -B), the 10-base logarithm value (Eq. (2)-C), and the constant multiplication value (Eq. (2)-D). The fact that $M \cdot 2^E$ form uses fewer bits in M and E, compared to the wordlength of the original value [15], means that the memory capacity associated with Eq. (2)-A/B/D can be significantly reduced. Item iii) can be formulated using Eq. (3) based on a set of given gain and $DR_{dB}$ values. Thus, it is possible to complete operation iii) using an RAM table to pre-store 8-bit gray-scale values mapped by the gain/$DR_{dB}$ setting in Eq. (3). In our plan, the maximum gain is 8 dB and $DR_{dB}$ below the 0-dB level ranges from -16 to -64 dB. Considering the resolution required to map 8-bit gray-scale images, the fractional accuracy of the logarithm value (Eq. (2)-C) is not particularly high, and therefore the capacity of the table related to Eq. (2)-C and Eq. (3) is also maintained at a reasonable level. Thus, the proposed architecture is a global memory-based display processing engine in which a lookup table is applied via operational steps i)-iii) mentioned above. As shown in Fig. 6, a scaling operation is used to globally reduce the number of redundant sign extension bits at the input in order to achieve an efficient wordlength. In our design, the maximum check value for E is 10, and the number of bits for M and E is 12 and 4, respectively. The wordlength distribution and memory size for each operation stage are shown in Fig. 6.

As mentioned above, the proposed two-stream RFFT computation unit can be used to achieve high A-line throughput based on two scenarios (dual A-line signal paths or a single high-rate path). The proposed display processing engine can complete the entire post-FFT operation to generate gray-scale data; therefore, the back-end display platform can produce the OCT image
directly without the need for further operations. Thus, the proposed system can be used to support to OCT systems with a high imaging rate. Table I lists the achievable performance of the proposed system under an assumed clock rate of 80 MHz.

\[
\begin{align*}
  a &= Ma' \times 2^{-Ea} = Ma \times 2^{-Eab} \\
  b &= Mb' \times 2^{-Eb} = Mb \times 2^{-Eab}
  \end{align*}
\]

\[
E_{ab} = \min\{Ea, Eb\}
\quad (1)
\]

\[
LC_{val} = 10 \times \log_{10} \left[ \text{nor.} \left( \frac{Ma^2}{(A)} + \frac{Mb^2}{(B)} \right) \times 2^{-E_{ab}} \right] \\
= 10 \times \log_{10} \left[ \text{nor.} \left( Ma^2 + Mb^2 \right) \right] - 10 \times \left( \frac{2E_{ab}}{\log_{10} 2} \right)
\quad (2)
\]

\[
\text{Gray Scale} = \text{round} \left\{ \left[ (LC_{val} + \text{Gain}) \times \left( \frac{255}{DR_{dB}} \right) \right] + 255 \right\}
\quad (3)
\]

Fig. 6. Block diagram showing architecture of the proposed global memory-based display processing engine.

| Number of FFT Points | Max. Axial (A-Line) Throughput \([\text{Single Optical Path}, \text{Dual Optical Paths}]\) | Max. Frame Rate \([\text{Resolution - axial x lateral}]\) |
|----------------------|-------------------------------------------------|-----------------------------------------------|
| 512                  | \(~312\ \text{KHz}, ~156\ \text{KHz} x 2\)       | \(~312\ \text{fps} (256 x 1000)\)              |
| 1024                 | \(~156\ \text{KHz}, ~78\ \text{KHz} x 2\)       | \(~156\ \text{fps} (512 x 1000)\)              |
| 2048                 | \(~78\ \text{KHz}, ~39\ \text{KHz} x 2\)        | \(~78\ \text{fps} (1024 x 1000)\)              |

4 Design Implementation and Verification

The proposed FDOCT image formation processor was subjected to system-level verification by implementing a test circuit on an FPGA platform (Xilinx /Spartan-6 module), as illustrated in Fig. 7. The FPGA platform communicated with a mobile phone through a USB communication. A set of raw
data (1000 lines of 512-sample A-line signal) captured from the FDOCT optical-mechanical front-end modules was pre-stored in the mobile phone. Verification was performed by first sending the data held in RAM and several streams (e.g. eight streams) of raw A-line data from the mobile phone to the FPGA platform via USB to be stored in RAM (for distribution by the re-sampler, DC removal, and display processing) and an input buffer (i.e., In_Buffer in Fig. 7). Following the operations of the FDOCT imaging processor (512-point RFFT; 48MHz clock rate) in the FPGA, 8-bit gray-scale data was sent back to the mobile phone via the USB link to display the OCT image. Following iterative application (e.g. 125 iterations based on eight A-line streams processed per iteration) of the above-mentioned operation, an OCT image is displayed on the screen of the mobile phone within a frame of 256 (axial) x 1000 (lateral). Another operating mode can then be applied to the FPGA circuit in order to perform FDOCT imaging operations for continuous A-line signal streams. In this mode, we were able to generate 256x1000 OCT images at 30 fps under a clock rate of 48 MHz. As shown in Table I, this is far from the highest performance that could be achieved using the proposed design. This less-than-ideal performance can be attributed to the relatively slow data-access mechanism (including the employment of a clock rate of just 48 MHz) used for the USB FIFO and display software. As shown in Fig. 7, the same raw data was used for Matlab-based simulations involving floating-point FDOCT imaging operations using the built-in FFT function in which display processing was achieved using direct arithmetic calculations (Eq. (3)). The OCT image obtained using FPGA is similar to that generated in Matlab, as shown in Fig. 7.

![Fig. 7. System-level verification of proposed FDOCT imaging processor based on FPGA platform and mobile phone display.](image-url)
5 Conclusion

In this paper, we report on a full FDOCT image formation processor for portable OCT applications. A hardware-efficient CFFT-based RFFT unit is presented to simultaneously process two streams of FDOCT imaging operations. Also, the proposed global memory-based display processing engine is capable of generating 8-bit gray-scale data for the direct and convenient display of images. Using the proposed FDOCT imaging processor, a high OCT image formation rate can be achieved. Furthermore, a system-level experiment based on an FPGA platform and a mobile phone verified the efficacy of the proposed design.