Real-time Software Implementation of Coherent Receiver DSP Adopting Multiplication-Based Parallel Frequency Offset Compensation for Fully Virtualized Access Networks

Takahiro Suzuki, Sang-Yuep Kim, Jun-ichi Kani and Tomoaki Yoshida

NTT Access Network Service Systems Laboratories, NTT Corporation, 1-1 Hikari-no-oka, Yokosuka-Shi, Kanagawa, 239-0847 Japan

takahiro.suzuki.pc@hco.ntt.co.jp

Abstract: This paper proposes multiplication-based parallel frequency offset compensation (FOC) for virtualized access networks and demonstrates real-time softwarization of the full DSP function suite, including polarization demultiplexing, FOC, carrier phase recovery, and decoding, on a server. © 2022 The Author(s)

1. Introduction

As traditional optical access systems depend on application specific integrated circuits (ASICs), they have great difficulty in meeting the fast-changing customer demands due to the long development cycles and high initial cost of the AISCs. To overcome this, network function virtualization (NFV) and software-defined network (SDN) have been actively promoted by telecom operators and vendors. The representative product is SDN-enabled broadband access (SEBA) [1], which enables control of access devices via common protocols by abstracting various types of optical line terminals (OLTs) and optical network units (ONUs). Studies to date focus on how a controller on a general-purpose server can manage the ASIC-based access system, whereas maximizing software components has been studied in order to reduce development cycle and capital expenditure (CAPEX) for flexible service creation. For dynamic bandwidth allocation (DBA) and physical coding sublayer (PCS) OLT functions, a 10-Gb/s software implementation has already been demonstrated on a general-purpose server [2, 3]. For future access networks using coherent detection, digital signal processing (DSP) softwarization continues to be studied as well. Our works have already demonstrated the DSP softwarization of fixed finite impulse response (FIR) filter, carrier phase recovery (CPR), and decoding using graphics processing units (GPUs) on a general-purpose server [4, 5]. Other works have tried to softwarize DSP functions [6,7], but no study has demonstrated DSP functions combined with polarization demultiplexing and frequency offset compensation (FOC). Especially, conventional FOC employs serial processing to calculate the total amount of phase changes, which makes high throughput processing difficult.

In this paper, we demonstrate real-time software DSP function suite, including polarization demultiplexing, FOC, CPR, and decoding, on a general-purpose server by proposing multiplication-based parallel FOC for the first time. Our FOC implementation significantly reduces GPU computation time by replacing the summation of frequency offset with multiplication; 5-Gb/s demonstration is achieved for polarization-multiplexed signals.

2. Coherent receiver DSP adopting multiplication-based parallel FOC

Figure 1 depicts a polarization-multiplexed coherent communication system, where the coherent receiver DSP is composed of a general-purpose server with GPU. Dual polarization-quadrature phase shift keying (DP-QPSK) signals are generated, and the optical signal is modulated orthogonal to the X and Y polarization in the transmitter side. The received signal is impacted by polarization interference, and the constellation rotates due to differences in the frequency between the local oscillator (LO) and the received signals. The GPU on the server runs DSP including polarization demultiplexing, FOC, CPR symbol decision, differential decoding, and Gray decoding.

Two implementation methods have been investigated for FOC. Figure 1(a) shows a typical FOC implementation with widely-used Viterbi-and-Viterbi algorithm on field-programmable gate arrays (FPGAs) or application specific integrated circuits (ASICs). The *k*-th signal input is represented as $\mathbf{r}(k)$. After multiplying $\mathbf{r}(k)$ by $\mathbf{r}(k-1)$, *m*-th power is used to calculate the phase difference, frequency offset $\Delta \varphi_k$, between consecutive symbols. This calculation can be parallelized for *N* symbols because it is independent for consecutive symbols. Given that the frequency offset occurs per symbol, the phase of *k*-th symbol is compensated by $\varphi(k) = \sum_{i=1}^{k} \Delta \varphi_i$. Thus, this summation must be serially implemented. Per *N* symbols, summation of frequency offset, $\varphi(k+N-1)$, is buffered. Since clock frequency of commercial FPGAs or ASICs is high, *N* values of 100 or less are sufficient for 10 Gb/s-class systems, and serial operations are possible. However, if we attempt software implementation, *N* must be larger because the short data transfers imposed by memory operations by both CPUs and GPUs reduce throughput. Therefore, a method is required that reduces the FOC processing time with large *N* to take advantage of GPU resources like multiple threads.

Figure 1(b) depicts the proposed multiplication-based parallel FOC implementation. This implementation calculates only one frequency offset $\Delta \varphi_k$ per N symbols, and the phase is compensated with $\varphi(k-1) + i\Delta \varphi_k$ for





 $\mathbf{r}(k+i)$ on N threads. Thread *i* multiplies *i* by $\Delta \varphi_k$ to estimate phase change by frequency offset, which reduces the serial summation processing and enables parallel processing of phase compensation after frequency offset calculation; moreover, the number of *m*-th power processing steps is reduced to 1/N. $N\Delta\varphi$ is buffered for calculating the subsequent N-symbol cycle frequency offset.

3. Performance evaluation of DSP with the proposed FOC

We conducted simulations and a real-time experiment to evaluate DSP performance with the proposed FOC; the experimental setup is shown in Fig. 1. A general-purpose server with Intel Xeon E5-2699v4 CPUs and a NVIDIA Tesla A100 GPU was used. The periodic 134 Mbyte data input is stored in memory and softwarized DSP was iterated. The IF card was an FPGA (Xilinx DK-V7-VC709-G) with four 1.25-GS/s analog to digital converter (ADC) modules with 8-bit resolution (4DSP FMC125). In the optical system, a distributed feedback (DFB) laser diode (LD) with 80-kHz linewidth was driven as a transmitter at approximately 1553 nm. 1.25-Gsymbol/s DP-QPSK signals with launched powers of -4.45 dBm were generated by an arbitrary waveform generator (AWG) programmed with the results of differential and Gray encoding of PRBS 23 and PRBS 17 for X and Y polarization, respectively; these signals were passed to an optical IQ modulator. The received optical power was adjusted by a variable optical attenuator (VOA) via 20-km single mode fiber (SMF) and input to the coherent optical receiver (Fujitsu FIM24723EB). The LO was the same type of LD used in the transmitter with 14-dBm power. IQ signals for each of the X and Y polarizations were input to the ADC module of the server, and the 5-Gb/s serial output of the server was input to an error detector (ED). A Monte Carlo (MC) BER simulation was carried out by executing the same DSP for the PRBS signals with frequency offset.

Figure 2(a) shows the frequency offset tolerance for each method. Without FOC, the frequency offset tolerance is very narrow. The theoretical estimation range of frequency offset is $[-Br/2m, Br/2m] \simeq$ [-156 (MHz), 156 (MHz)], where *Br* is the symbol rate and *m* is the number of constellation states. The simulation and real-time experimental results show that the proposed FOC successfully compensates the frequency offset basically within the range of the theoretical values. Figure 2(b) shows the bit error rate (BER) performance for each method. When the frequency offset (FO) is 100 MHz, demodulation failed without FOC. Whereas, with our proposal, the softwarization is achieved. This result shows that our FOC with 100 MHz of frequency offset has almost the same BER performance with the case with zero FO. In addition, similar performance was obtained using a 20 km fiber. Figure 2(c) shows the processing time of the method that applies the conventional implementation to software. Due to its serial summation shown in Section 2, its processing time is excessive. The processing time of the proposed implementation is shown in Fig. 2(d). The proposed implementation reduces the processing time by a factor of



Fig. 2. (a) frequency offset tolerance, (b) BER performance, (c) processing time of the conventional implementation, and (d) processing time of the proposed implementation.

1703. The GPU transfers data in 134-Mbyte units for 4 channels and executes DSP. In order to execute 5 Gb/s demodulation for 40 Gb/s signal input (i.e. 1.25-GS/s sampling rate, 4 channels and 8 bit vertical resolution), DSP execution must be complete within 26.8 ms. The result shows that the processing time satisfies the constraint time and our proposal achieves 5-Gb/s DSP softwarization.

4. Conclusion

This paper demonstrated 5 Gb/s DSP softwarization for fully virtualized access networks by proposing FOC software implementation that replaces the summation of frequency offset with multiplication of sampling number. A real-time experiment showed that it was about 1703-times faster than the conventional implementation while maintaining the BER performance.

References

- 1. S. Das, "From CORD to SDN Enabled Broadband Access (SEBA) [Invited Tutorial]," in Journal of Optical Communications and Networking, vol. 13, no. 1, pp. A88-A99, Jan. 2021.
- F. Slyne, et al., "Experimental Demonstration of multiple Disaggregated OLTs with Virtualized Multi Tenant DBA, over General Purpose Processor," OFC, M3Z.11, 2020.
- T. Suzuki, et al., "Demonstration of Fully Softwarized 10G-EPON PHY Processing on A General-Purpose Server for Flexible Access Systems", Journal of Lightwave Technology, vol. 38, Issue 4, pp. 777-783, Feb. 2020.
- S. -Y. Kim, et al., "Demonstration of Real-time Coherent 10-Gb/s QPSK Reception Implemented on a Commodity Server," 2021 European Conference on Optical Communication (ECOC), pp. 1-4, 2021.
- T. Suzuki, et al., "Real-Time Implementation of Coherent Receiver DSP Adopting Stream Split Assignment on GPU for Flexible Optical Access Systems", IEEE/OSA Journal of Lightwave Technology, Vol. 38, Issue 3, pp. 668-675, Oct. 2019.
- S. van der Heide, et al., "Real-Time 10,000 km Straight-Line Transmission Using a Software-Defined GPU-Based Receiver," in IEEE Photonics Technology Letters, vol. 33, no. 24, pp. 1519-1522, Dec. 2021.
- S. van der Heide, et al., "Field Trial of a Flexible Real-Time Software-Defined GPU-Based Optical Receiver," J. Lightwave Technol. 39, 2358-2367 (2021).