Parallel Timing Synchronization Algorithm and Its Implementation in High Speed Wireless Communication Systems

Xin Hao^{*†}, Qiuyu Wu^{*†}, Zhaohui Wang^{*†}, Jin Fan^{*†}, Ying Wang^{*†} and Changxing Lin^{*†} *Institute of Electronic Engineering, China Academy of Engineering Physics, Mianyang, 621900 China [†]Microsystem & Terahertz Research Center, China Academy of Engineering Physics, Chengdu, 610200 China E-mail: linchangxing@mtrc.ac.cn Tel: +86-816-2497242

Abstract—For the sake of high speed wireless communication, this paper proposes a high speed parallel digital timing synchronization algorithm. The algorithm is a joint scheme composes of index-associated rearrangement based parallel FIFOs and a serial digital timing recovery based dual feedback loop. The proposed parallel FIFOs saves 67% LUTs of traditional method. In the feedback loop, to further reduce the complexity, a direct-calculation parallel NCO structure is proposed; meanwhile, to guarantee the high accuracy, an all effective timing errors contained loop filter is proposed. To validate the algorithm, float and fix point simulations were performed on MATLAB platform. The performance loss introduced are less than 0.5 dB and 1 dB respectively, which confirmed the high efficiency of the algorithm. Furthermore, the algorithm is implemented on a Xilinx XC7VX485T FPGA chip, a 20 giga bit per second (Gbps) online 16QAM timing synchronization is achieved at the running frequency of 159.524MHz with two times oversampling.

I. INTRODUCTION

With the increasing demand of transmission capacity and the limitation of hardware processing rate, parallel digital signal processing in very high speed wireless communication systems [1-3] has attracted a large amount of interest in recent years. Therefore, the popularity of high throughput wireless communications [4] has put forward higher requirements for the implementation of parallel digital signal processing algorithms.

Baseband digital signal processing (DSP) is indispensable in wireless communication systems, meanwhile synchronization is an imperative block in baseband DSP, in which timing synchronization is required to eliminate timing errors. Parallel digital timing synchronization is a powerful method to achieve high capacity in wireless communication systems.

Most parallel algorithms are deducted from serial algorithms. There are two basic types of serial timing synchronization algorithms: feed-forward timing synchronization, O&M algorithm [5] as an example; and feed-back timing synchronization, e.g. Gardner's algorithm [6-8].

In [9], the authors have successfully implemented a parallel O&M algorithm in FPGA. However, in O&M algorithm, a four times oversampling is ineluctable, this will lead to a mass hardware resource consumption, which increases the system cost. To overcome this shortage, Zhou and her collaborates

have developed a parallel timing synchronization algorithm [10-11], which is based on Gardner's algorithm with two times oversampling. Nevertheless, Zhou's algorithm ignores some error information in loop filter, thus leads a greater deviation in recovered signals; meanwhile, the algorithm is verified offline, which leads insatiable for real time communications such as HD video calling.

In this paper, we presented a parallel timing synchronization algorithm with two times oversampling rate, and the feasibility of the algorithm is verified with a XC7VX485T FPGA chip.

This paper is organized as follows: Section II describes traditional serial timing synchronization algorithm. The proposed algorithm is carried out in Section III. Section IV presents the simulation and implementation results. Finally, a conclusion is made in Section V.

II. SERIAL TIMING SYNCHRONIZATION ALGORITHM

The proposed algorithm is based on Gardner timing synchronization algorithm, and two times oversampling sample time relations are shown in Fig. 1. Where, μ_k stands for a fractional interval, m_k is defined as a basepoint index. T_s is the sampling period of serial analog-to-digital conversion (ADC), T_i is the synchronized sample time, and k is the index of T_i .

The Structure of Gardner algorithm is described by Fig. 2. In the structure, there are four submodules, the interpolator, the



Fig. 1 Two Times Oversampling Serial Timing Synchronization Sample Time Relations



Fig. 2 Gardner Timing Synchronization Loop

timing error detector (TED), the loop filter and the numbercontrolled oscillator (NCO).

A. Interpolator

In a serial system, the authors have summarized [7], 3 multipliers or dividers will be consumed with cubic interpolator; and 2 with four-point piecewise-parabolic interpolator (α =1/2). In a parallel system, it will increase with parallel quantities. Take a 64-parallel system for instance, 384 multipliers/dividers will be consumed with cubic interpolator; and 256 with four-point piecewise-parabolic interpolator. In section III, a non-multiplier/divider consumption interpolator based on four-point piecewise-parabolic will be proposed. The coefficients for four-point piecewise-parabolic (α =1/2) is shown in equation (1).

$$C_{-2} = \frac{1}{2} \cdot x^{2} - \frac{1}{2} \cdot x$$

$$C_{-1} = -\frac{1}{2} \cdot x^{2} + \frac{3}{2} \cdot x$$

$$C_{0} = -\frac{1}{2} \cdot x^{2} - \frac{1}{2} \cdot x + 1$$

$$C_{1} = \frac{1}{2} \cdot x^{2} - \frac{1}{2} \cdot x$$
(1)

B. Timing Error Detector

Gardner's timing error detector [5] is employed, as is described in equation (2)

$$e(n) = I\left(n - \frac{1}{2}\right)[I(n) - I(n-1)] + Q\left(n - \frac{1}{2}\right)[Q(n) - Q(n-1)]$$
(2)

Where, e(n) is the timing error, I(n) and Q(n) are real and image part output of the interpolators, n - 1, n - 1/2, n are three continuous indexes.

C. Loop Filter

The output of timing error detector serves as the input of loop filter, which is responsible for filtering the residual noise of



Fig. 3 Structure of Proportional Integral Filter

timing error detector. The most commonly used filter is

proportional integral filter, which is composed of a proportional unit and an integral unit, its structure is shown in Fig. 3.

D. NCO

As μ_k and m_k are both provided by NCO, it is an extremely important module. The control word of loop filter is defined as W(n), which serves as the input of NCO, then the difference equation is

$$\eta(m) = [\eta(m-1) - W(m-1)] \mod 1$$
(3)

The NCO relations are shown in Fig. 4. From similar triangles, it can be seen that



Fig. 4 NCO Relations

$$\frac{\mu_k T_s}{\eta(m_k)} = \frac{(1-\mu_k)T_s}{1-\eta(m_k+1)}$$
(4)

 μ_k can be solved as

$$\mu_k = \frac{\eta(m_k)}{1 - \eta(m_k + 1) + \eta(m_k)} = \frac{\eta(m_k)}{W(m_k)}$$
(5)

As $1/W(m_k) \approx T_i/T_s$, then

$$\mu_k = \varepsilon_0 \cdot \eta(m_k) \tag{6}$$

Where, $\varepsilon_0 = 1/W(m_k)$, which means μ_k can be achieved by a single shift operation on a FPGA chip with integral multiple sampling.

III. THE PROPOSED ALGORITHM

The proposed parallel timing synchronization algorithm is composed of index-associated rearrangement based parallel FIFOs and a serial digital timing recovery based dual feedback loop. Fig. 5 depicts the 2m-parallel structure of the proposed algorithm. First, the analog signals are sampled by 2 high speed ADCs (sample rate $2m \cdot f_s$) for generating parallel digital signals. To ensure steady data flow, the digital signals are saved and rearranged by parallel FIFOs. Afterwards, renewed signals will be achieved by interpolators and imported to TEDs to obtain timing errors which will be filtered by loop filter. Ultimately, timing errors can be compensated with fractional interval and basepoint index provided by NCOs.



Fig. 5 2m-Parallel Structure of Proposed Algorithm

A. Parallel FIFOs

Rearrangement is critical to signals in parallel timing synchronization, parallel FIFOs based delete-keep algorithm [12] is fit for rearranging data. However, in [12], the authors took an index-independent method, which leads a continuous change of the subscript of FIFOs. Take 64-parallel structure for instance, a 64x64 lookup table (LUT) is required. The LUT

TABLE I. INDEX RELATION

independent index	associated index				
index(i)	index(i)				
index(i + 1)	index(i)+1				
:	:				
index(i + 63)	index(i)+63				

Re	source	LUT	FF	
I Idilianti an	independent	21248	3175	
Ounzation	associated	6881	2531	
Litilization 0/	independent	7.00	0.52	
Othization%	associated	2.27	0.42	
Ava	ailable	303600	607200	

TABLE II. RESOURCE UTILIZATION COMPARISON

consumption will increase exponentially with parallelism. To cope with this problem, an index-associated method is proposed.

The "delete-keep" operation is controlled by the basepoint index m_k of the last NCO depicted in Fig. 5. Actually, in data rearrange module, the indexes increase only 1 between adjacent FIFOs. The proposed associated index algorithm is exhibited in Table I.

As is shown in Table I, the index numbers in the independent method are variable values, which leads to searching operation in the LUTs every parallel cycle. However, it is easy to find out that the index value and the of the index number value are the same. Based on this theory, the increment of index number can be translated to index value directly. This is the associated index method. With the associated index method, only one single searching operation need to be conducted during one parallel cycle, and the other searching operations conducted in independent index method are translated to add operations. Take 64 parallel for instance, the LUT searching will be carried out in index(i), and the other 63 searching operation in independent index method are replaced with adders.

The resource utilization results after synthesis of parallel FIFOs of independent and associate method is shown in table. II, which shows that the proposed index-associated method saves about 67% LUTs of the index-independent method adopted in [12].

In order to save LUTs, data-select module uses similar technology as in data rearrange module. Furthermore, another

four data belong to the next clock cycle are added to the rearranged queues as each interpolator needs four data every time.

B. Parallel Gardner Loop

The parallel Gardner loop is based on PDTRL algorithm [13]. It is composed of parallel modules and loop filter. Nevertheless, the NCO and loop filter are achieved in much briefer and more accurate way.

a) Parallel Module

Each parallel module (PM) has two interpolators, one timing error detector (TED) and one NCO.



Fig. 6 Farrow Structure for Piecewise-Parabolic Interpolator (a=1/2)

The coefficients of four-point piecewise-parabolic interpolator with $\alpha=1/2$ are all integral multiple of 2, which can be implemented with shift operations on FPGA. To reduce hardware source consumption, a multiplier/divider free interpolator is proposed based on aforementioned technology. If " $3/2 \cdot x$ " of C_{-1} in equation (1) is separated into " $1/2 \cdot x + x$ ". Then a four-point piecewise-parabolic interpolator can be implemented on a FPGA chip with only shift operations and adders/subtractors. A multi-free four-point piecewise-parabolic interpolator will be employed in the proposed parallel algorithm in order to take a balance between resource consumption and computational accuracy. The farrow structure of four-point piecewise-parabolic is shown in Fig. 6.

TED equation is shown below

$$e(n,i) = I_1(n,i)[I_2(n,i) - I_2(n,i-1)] +Q_1(n,i)[Q_2(n,i) - Q_2(n,i-1)]$$
(7)

Where, e(n, i) is the timing error of the *i*th PM at time n, $I_1(n, i)$ is the first interpolator's output of the *i*th PM at time n, $I_2(n, i - 1)$ is the second interpolator's output of the *i*th PM at time n. When i = 1, i - 1 stands for the second interpolator's output of the last PM at time n - 1. And road Q has the same explanation.

In [13] the authors proposed correlative calculation method to achieve parallel NCO control. The overflow is achieved by a comparator. Nevertheless, it is logic complicated and resource consuming. To reduce complexity and save resource, this paper proposed a direct-calculation method to achieve the overflow moment. Gardner gives equation below in [6] to calculate m_k

$$m_k = \inf[k T_i / T_s] \tag{8}$$

Where, int[z] means largest integer not exceeding z, T_s is the serial sample time, T_i is the synchronized sample time. Eq. (8) can be translated into Eq. (9) shown below

$$m_{k+1} = m_k + \text{fix}[R + W(n) + \mu_k]$$
 (9)

Where, fix[z] stands for the largest integer toward z, and R is half of oversampling rate, W(n) is the control word of NCO at time n. Thus, m_k of each parallel module can be calculated directly and accurately without complex comparison logic.



Fig. 7 Structure of Adopted Loop Filter

b) Loop Filter

A proportional integral filter is employed. In [13], the timing error of the last TED served as the input of proportional element. Thus, the information carried by the other TEDs are ignored, which will lead a greater deviation. To guarantee the effectiveness of all timing errors, the average value of them is employed as the input of loop filter in this paper. Simulation results confirmed that the performance is better when using the average value instead of the last value.

Proportional element

$$P_n = k_1 \times (err_{n,1} + err_{n,2} + \dots + err_{n,32})$$
(9)

Integral element

$$I_n = I_{n-1} + k_2 \times \left(err_{n,1} + err_{n,2} + \dots + err_{n,32} \right)$$
(10)

Then the output of loop filter is

V

$$V_n = P_n + I_n \tag{11}$$

Where, k_1 and k_2 stand for the coefficient of proportional element and integral element separately, $err_{n,i}$ is the error of the *i*th PM at time *n*, W_n is the output of loop filter at time *n*. The structure is represented in Fig. 7.

IV. SIMULATION AND IMPLEMENTATION RESULTS

The proposed algorithm is investigated in a baseband communication system. In this case, modulation type is 16QAM, bit rate is 20Gbps, roll-off factor is 0.4, oversampling rate is 2 times of symbol rate ($f_s = 2R_s$), and the timing frequency and phase offset are 32kHz and π respectively. A 64-parallel structure is adopted to purchase 20Gbps bit rate. The received signal is quantized to 8 bits in fix point simulation and FPGA.

The performance of the proposed parallel algorithm is simulated in MATLAB. The VHDL code version of this algorithm is simulated in Vivado with the same source in MATLAB.

A. MATLAB Simulation



(a)Before Timing Synchronization (b)after Timing Synchronization



Fig. 8 Constellation Diagrams

Fig. 9 Performance of the Proposed Algorithm

Fix point simulation constellation diagram before and after the timing synchronization in the baseband system is shown in Fig. 8(a) and Fig. 8(b). In this simulation, SNR is set to 20dB.

The bit error rate of proposed algorithm is carried out in Fig. 9. It indicates that the algorithm can work efficiently with performance loss less of float and fix point simulation are 0.5dB and 1 dB respectively.

B. FPGA Implementation

Hardware implementation is carried out on Xilinx XC7VX485T FPGA chip. 128 parallel ROMs are employed to store the source data as an equivalent substitution of two 10GHz ADCs. In order to make the comparison between MATLAB and FPGA, periodic source is embedded into the ROMs. The write clock and read clock of parallel FIFOs module are set to 156.268MHz and 159.524MHz respectively by a MMCM. The device utilization is summarized in table III.

Fig. 11(a) shows the NCO output of fix point simulation on MATLAB platform and behavior simulation with ROMs on Vivado platform. The "*" represents MATLAB value and the " Δ " represents VHDL value. It can be seen that the two curves are totally overlapped. Fig. 11(b) exhibits the difference value of NCO output achieved on MATLAB and VHDL. The curve value is constantly 0, which represents the NCO output in MATLAB is exactly the same as in VHDL. Moreover,

simulation results show that all the values achieved in VHDL is exactly the same as those in MATLAB, which confirmed the correctness of behavior simulation.

The constellation diagram of FPGA implementation is displayed in Fig.12(a). And Fig. 12(b) shows the difference

Resource	Utilization	Available	Utilization%		
LUT	35228	303600	11.60		
LUTRAM	1375	130800	1.05		
FF	55950	607200	9.21		
BRAM	143	1030	13.88		
DSP	480	2800	17.14		

TABLE III. DEVICE UTILIZATION SUMMARY

	value ((image	part)	of	interpolator's	output	of	MATL	AB	fix
--	---------	--------	-------	----	----------------	--------	----	------	----	-----



Fig. 11 NCO Output of Fix Point MATLAB Simulation and FPGA Implementation



Fig. 12 Constellation Diagram and Difference Value (Image Part) of Interpolator's Output with MATLAB Fix Point Simulation and FPGA Implementation

point simulation and implementation. As the initial values are unable to capture on a FPGA chip, the first 15000 difference values are meaningless in Fig. 12(b), and the difference value is always less than 3 from about the 15000th data, which confirms that the implementation of the proposed algorithm can work correctly on a FPGA chip when the algorithm converges.

V. CONCLUSIONS

In this paper, we have proposed a novel parallel digital timing synchronization algorithm and its implementation structure. Simulations in MATLAB and Vivado demonstrate the effectiveness and feasibility in high speed wireless communication systems. Moreover, a 20 Gbps online implementation with 16QAM modulation is achieved on XC7VX485T FPGA, and the results reveal high consistency when comparing with simulation. Furthermore, the proposed algorithm is not limited to 64-parallel, a higher bit rate can be achieved with faster clocks or more parallels.

ACKNOWLEDGEMENT

This paper was supported by Defense Industrial Technology Development Program, No. JCKY2016212C045.

REFERENCES

- [1] T. Yang, C. Shi, X. Chen, M. Zhang, Y. Ji, F. Hua, and Y. Chen, "Linewidth-tolerant and multi-format carrier phase estimation schemes for coherent optical m-QAM flexible transmission systems," Optics Express Vol. 26,Issue 8,pp. 10599-10615, 2018
- [2] Z. Gao, M. Zhou, P. Reviriego and J. A. Maestro, "Efficient Fault-Tolerant Design for Parallel Matched Filters," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 65, no. 3, pp. 366-370, March 2018.
- [3] C. Lin B. Shao and J. Zhang "A high data rate parallel demodulator suited to FPGA implementation " in ISPACS 2010-2010 International Symposium on Intelligent Signal Processing and Communication Systems Proceedings 2010.
- [4] Q. Wu et al., "A 21 km 5 Gbps real time wireless communication system at 0.14 THz," 2017 42nd International Conference on Infrared, Millimeter, and Terahertz Waves (IRMMW-THz), Cancun, 2017, pp. 1-2.
- [5] M. Oerder and H. Meyr, "Digital filter and square timing recovery," in IEEE Transactions on Communications, vol. 36, no. 5, pp. 605-612, May 1988.
- [6] F. M. Gardner "Interpolation in digital modems part I: Fundamentals "IEEE Trans. Commun. vol. 41 no. 3 pp. 501-507 1993.
- [7] L. Erup F. M. Gardner and R. A. Harris "Interpolation in digital modems - part II: Implementation and performance " IEEE Trans. Commun. vol. 41 no. 6 pp. 998-1008 1993.
- [8] F. Gardner, "A BPSK/QPSK Timing-Error Detector for Sampled Receivers," in IEEE Transactions on Communications, vol. 34, no. 5, pp. 423-429, May 1986.
- [9] C. Lin, J Zhang, and B. Shao, "A High Speed Parallel Timing Recovery Algorithm and its FPGA implementation," 2011 International Symposium on Intelligence Information Processing and Trusted Computing, 63-66. Oct. 2011.
- [10] X. Zhou, X. Chen, W. Zhou, Y. Fan, H. Zhu and Z. Li, "All-Digital Timing Recovery and Adaptive Equalization for 112 Gbit/s POLMUX-NRZ-DQPSK Optical Coherent Receivers," in IEEE/OSA Journal of Optical Communications and Networking, vol. 2, no. 11, pp. 984-990, November 2010.
- [11] Y. Fan, X. Chen, W. Zhou and X. Zhou, "Parallel processing clock synchronization-dispersion equalization combining loop in 112Gb/s optical coherent receivers," The 19th Annual Wireless and Optical Communications Conference (WOCC 2010), Shanghai, 2010, pp. 1-4.
- [12] C Lin, J Zhang, B Shao, "A Multi-Gigabit Parallel Demodulator and Its FPGA Implementation," IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences, vol 95, 1412-1415, 2012

[13] X. Zhou, X. Chen, "Parallel implementation of all-digital timing recovery for high-speed and real-time optical coherent receivers," Optics Express, vol 19, 9282-9295, 2011