Efficient Coarse Frequency Synchronizer Using Serial Correlator for DVB-S2

Jang Woong Park, Hyoung Jin Yun,
Myung Hoon Sunwoo
School of Electrical and Computer Engineering
Ajou University
San 5, Wonchun-Dong, Yeungtong-Gu,
Suwon, 443-749 Korea
sunwoo@ajou.ac.kr

Pansoo Kim, and Dae-Ig Chang
Electronics and Telecommunications Research Institute
(ETRI), Broadband Wireless Multimedia Research Team
161, Gajeong-Dong, Yuseong-Gu,
Daejeon, 305-700 Korea
pskim@etri.re.kr

***Abstract*—This paper proposes an efficient coarse frequency synchronizer for digital video broadcasting – second generation (DVB-S2). The input signal requirement of acquisition range for coarse frequency estimator in the DVB-S2 is around ±1.5625Mhz, which corresponds to 6.25% of the symbol rate at 25Mbaud. At the process of analyzing the robust algorithm among data-aided approaches, we find that the Luise & Reggiannini (L&R) algorithm is the most promising one for coarse frequency estimation with respect to estimation performance and complexity. However, it requires many multipliers and adders to compute output values of correlators. We propose an efficient architecture identifying the serial correlator with the buffer and multiplexers. The proposed coarse frequency synchronizer can reduce the hardware complexity about 92% of the direct implementation. The proposed architecture has been implemented and verified on the Xilinx Virtex II FPGA.**

I. INTRODUCTION

Digital video broadcasting via satellite (DVB-S) was standardized in 1994 and adopted by most of satellite broadcasting service, today [1]. As a demand for high quality videos and interactive services increase, the service operators of satellite broadcasting needed to improve the spectral efficiency against the existing DVB-S system. As a result, the standardization work of DVB-S2 was completed in 2003 [2]. Its bandwidth efficiency was improved up to 30% in the same conditions with DVB-S since the powerful coding schemes, namely the low-density parity check (LDPC) codes, are utilized, together with high order modulation such as 8/16/32APSK.

However, the LDPC codes require the new frame structure and synchronization algorithms in order to operate at very low signal to noise ratio (SNR) environment compared with DVB-S. Moreover, the new architecture has to work with the existing outdoor equipment with the Low Noise Block-downconverter (LNB) for DVB-S. This implies it should be compatible with the same phase noise as specified for DVB-S [3][4]. Similarly, for the mass-commercial production, the receiver typically is developed with a low-cost local oscillator, which will lead to large initial frequency offset. In summary, the carrier synchronization must be the one of very significant parts to design the DVB-S2 demodulator.

The maximum initial frequency offset in the DVB-S2 is around ±5Mhz, which represents 20% of the symbol rate 25Mbaud. Some promising candidates among Data-Aided (DA) carrier frequency recovery algorithms can be used to cover these frequency errors [5] under the above strict requirement. To eliminate such large frequency offset, the frequency synchronizer is usually divided into 2 or 3 parts, such as initial, coarse, fine synchronizer according to its specific acquisition range, accuracy and noise immunity. In this paper, the overall frequency offset synchronization was specified with initial, coarse, and fine estimation and each maximum lock range corresponds to 20%, 6.25%, and 0.03%, respectively. Fig. 1 shows the proposed synchronizer.

In the previous work, we designed the efficient initial frequency synchronizer in [6]. This paper proposes the new coarse frequency synchronizer. To alleviate the frequency offset from 6.25% to below 0.03%, the Luise and Reggiannini (L&R) [7] algorithm was employed through verification process of computer simulations. The L&R algorithm can provide the proper hardware complexity and estimation accuracy performance compared with the Mengali and Moreli (M&M) [8] and Fitz [9] algorithms at the low SNR conidtion. Although the L&R algorithm is the best for coarse frequency synchronization, it still occupies large chip size because it requires a number of complex multiplications. To reduce the hardware complexity, we propose the new cost-effective coarse frequency synchronizer based on the L&R algorithm.

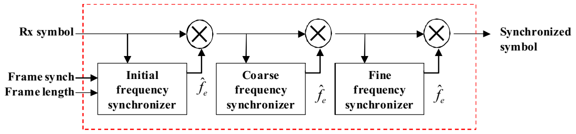
Rx symbol
Frame synch
Frame length
Initial frequency synchronizer
$\hat{f}_e$
Coarse frequency synchronizer
$\hat{f}_e$
Fine frequency synchronizer
$\hat{f}_e$
Synchronized symbol

Fig. 1. Proposed frequency synchronizer

This paper is organized as follows. Section II describes some kinds of DA algorithms, Section III presents the new

978-1-4244-1684-4/08/$25.00 ©2008 IEEE

cost-effective architecture of the L&R algorithm, and Section IV gives the performance comparisons with the direct implementation. Finally, conclusions are drawn in Section V.

II. Coarse Frequency Estimation Algorithms

In general, DA algorithms are used to meet satisfactory performance with known data identified as preambles and pilot symbols, easily. In DVB-S2, the unmodulated pilot symbols shown in Fig. 2 are available for frequency and phase estimations. In this section, we show the analysis among the three algorithms proposed by M&M[8], L&R[7] and Fitz[9], which are shown in Table I. Then, we can choose the most appropriate algorithm for the coarse frequency synchronizer through the simulations because each algorithm has different acquisition ranges, hardware complexity and noise immunity.

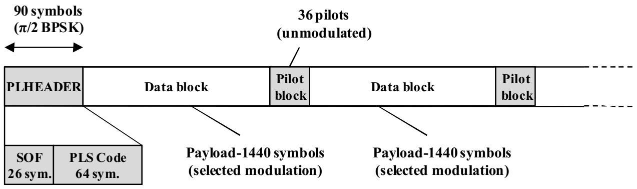

Fig. 2. DVB-S2 PLFRAME scheme [4]

First of all, the M&M algorithm is specified like the Eq.(1), which can achieve the frequency offset estimation by applying weighted average with arg$\{R_n(k)R^*(k\text{-}1)\}$, a phase difference between neighbor pilot symbols. $R_n(k)$ shown in the Eq.(5) is the correlation output between DA data samples, performed commonly in all of three algorithms, where L_p is the pilot block length, M is a design parameter not greater than $L_p/2$, $p^n{}_i$ is the i-th received pilot symbol of the n-th pilot block, and c_i is the referenced i-th pilot symbol. Although this algorithm has the large maximum acquisition range such as 1/2, the weighted average operation increases hardware complexity. Meanwhile, the L&R algorithm shown in the Eq.(3) and the Fitz algorithm shown in the Eq.(4) estimate the frequency offset using the phase difference of the complex sum from $R(1)$ to $R(M)$. As shown in equations, the difference between these algorithms is only the summation method after correlation process of Eq. (5). The maximum acquisition ranges of the L&R and the Fitz algorithm are equivalent to $1/(M+1)$ and $1/(2M)$, respectively.

TABLE I. Three Algorithms Proposed by M&M, L&R and Fitz

M&M [8]
$\hat{f}_{e,M\&M} = \frac{1}{2\pi T_s}\sum_{k=1}^{M} l_k \arg\{R_n(k)R_n^{\ *}(k-1)\}$ (1) $l_k = 3\frac{(L_p-1)(L_p-k+1)-M(L_p-M)}{M(4M^2-6ML_p+3L_p^{\ 2}-1)}$ (2)
L&R [7]
$\hat{f}_{e,L\&R} = \frac{1}{\pi T_s(M+1)}\arg\left\{\sum_{k=1}^{M} R_n(k)\right\}$ (3)
Fitz [9]
$\hat{f}_{e,Fitz} = \frac{2}{\pi T_s M(M+1)}\sum_{k=1}^{M}\arg\{R_n(k)\}$ (4)
Common Sample Correlations
$R_n(k) = \frac{1}{L_p-k}\sum_{i=k}^{L_p-1} p_i^n c_i^*(p_{i-k}^n c_{i-k}^*)^*, \quad 0 \le k \le M-1$ (5)

MATLAB simulations are performed and Fig. 3 shows the performance comparisons of the coarse frequency synchronizers based on three algorithms in the variation of M. Simulation parameters are fixed by E_s/N_o=-2.35dB, the maximum frequency offset=0.0025R_s, L_p=36, and the number of pilot block is 100. As shown in Fig. 3, the normalized standard deviations (STD) of frequency offset estimation error against the L&R and Fitz algorithms are converged over M=11 and the M&M algorithm is done over M=9. With empirical the results, we decide the variable M=11 of the L&R algorithm. Moreover, in the Fig. 3, the M&M algorithm is degraded compared with other two algorithms in terms of frequency STD performance.

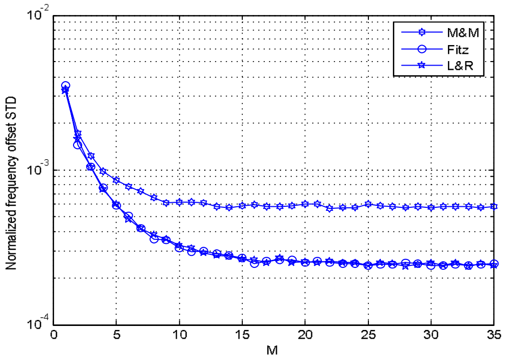

Fig. 3. Performance compariosons of three coarse frequency synchronizers in the variation of M

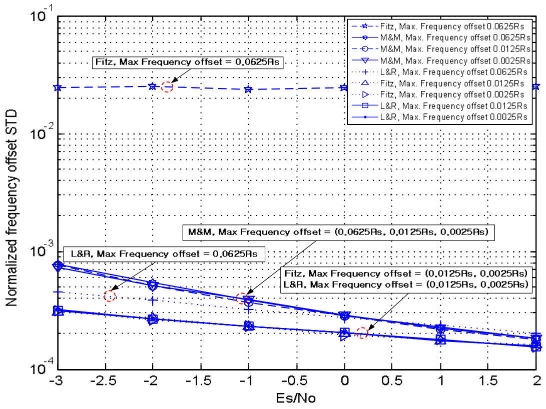

Fig. 4. Performance compariosons of three coarse frequency synchronizers in the variation of E_s/N_o and estimation range

Fig. 4 shows the performance comparisons of the coarse frequency synchronizers based on three algorithms with respect to E_s/N_o and estimation range. Simulation parameters are fixed by M=11 and the number of pilot block 100. The results show the L&R algorithm has the best performance at the conditions of 0.0625R_s and the low E_s/N_o. However, the Fitz algorithm shows the performance degradation because its estimation range is limited below $1/(2M)$, that is, 0.0385R_S. In addition, the M&M algorithm has the large estimation error at the low E_s/N_o compared with the L&R algorithm. Through the verification process, the L&R algorithm with M=11 is chosen to estimate the coarse frequency offset for DVB-S2, finally.

III. Proposed Hardware Architecture

Based on the L&R algorithm, we design the coarse frequency synchronizer. The frequency estimation of the L&R

algorithm is realized by accumulating the correlations of the known training sequence. To improve estimation performance, we accumulate the consecutive pilot blocks as many as L. Hence, the proposed frequency estimator updates its estimation value at every pilot block. The L&R frequency estimator with M=11 and L_p=36 is shown in the Eq.(6).

$$\hat{f}_{e,L\&R} = \tan^{-1}\left\{\sum_{n=1}^{L}\sum_{i=1}^{11}\left(\frac{1}{36-i}\sum_{k=i+1}^{36} p_i^n \times p_{i-k}^{n\;*}\right)\right\} \qquad (6)$$

Fig. 5 shows the basic architecture based on the Eq. (6) and Fig. 6 shows the detail circuits for each correlator. The number of correlators is 11, which is equivalent to the same architecture in [7].

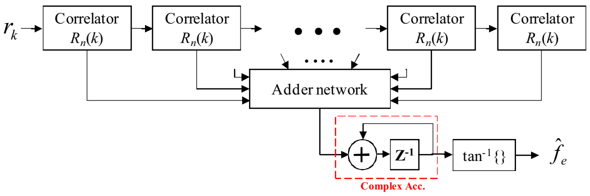

Fig. 5. Block diagram of the frequency synchronizer

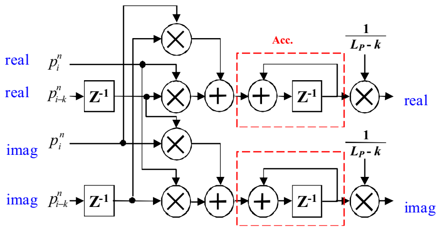

Fig. 6. Detail circuit of each correlator for $R_n(k)$

One complex multiplication is shown in the Eq. (7). The front multipliers and adders described in Fig. 6 accomplish the multiplications of Eq. (7). In addition, the registers are cascaded the next correlator, iteratively. The first summation of Eq. (6) is implemented through the accumulation process in Fig. 6 and the second is implemented by the adder network in Fig. 5. The last one is done by the accumulator in Fig. 5. Because each correlator for computing $R_n(k)$ consists of 6 real multipliers, 4 adders and 4 registers, the coarse frequency estimator with 11 correlators, consists of 66 multipliers, 44 adders and 44 registers. Moreover, the adder network and the accumulator require 22 adders and 2 registers since one complex accumulator is composed of 2 adders and 2 registers. Totally, 66 multipliers, 66 adders and 46 register are required to implement L&R frequency estimator.

$$p_i^n \times (p_{i-k}^n)^* = (a+jb)\times(c-jd) = (a\times c+b\times d)+j(-a\times d+b\times c) \qquad (7)$$

We find out the data width of the L&R frequency estimator has to be over 20 bits through the simulations. The implementation results in 66 multipliers with 20 bits that occupy huge silicon resource. Similarly, other computation units have the same difficulty. Therefore, the computation logic must be minimized to reduce the hardware complexity.

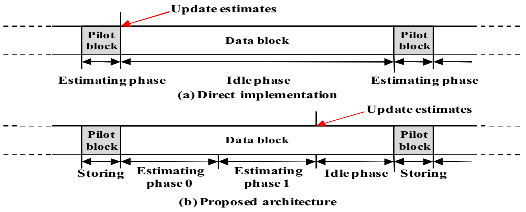

Fig. 7. Comparison between direct and proposed architecture

Fig. 7(a) shows the operation time of the existing frequency estimator which yields the new estimation value at every pilot block. In the direct implementation, the parallel correlators repeat shifting and computation during the pilot block and complete the frequency estimation when the last pilot symbol comes in. Then, the existing coarse frequency estimator keeps the idle phase (i.e. the value of the frequency estimator is kept constant) during the data block. This paper proposes the serial correlation architecture to utilize this idle phase for minimizing the hardware complexity.

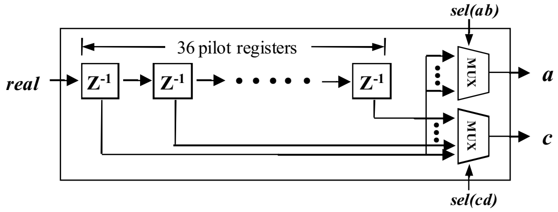

Fig. 8. Registers and multiplexers block

We divide the overall estimation process time into the storing phase, estimating phase 0, estimating phase 1 and idle phase as shown in Fig. 7(b). First, the 36 pilot symbols are kept in the shift register shown in Fig. 8 during the stored phase. After pilot symbols are stored, the real part of correlation is computed during the estimation phase 0. After the imaginary part is computed during the estimation phase 1, the frequency estimator yields the estimated frequency value. The frequency offset has almost uniform characteristics and many frequency estimations are continuously accumulated at every pilot block. Therefore, we can see the same result with the direct implementation though the estimation value is generated somewhat late. By using the proposed method, we can dramatically reduce the hardware complexity.

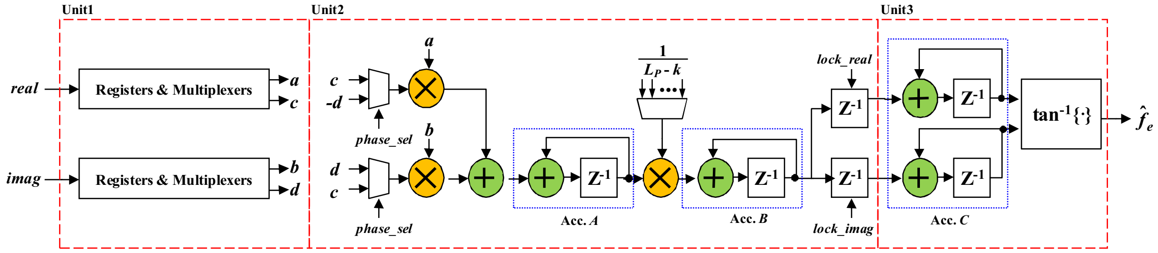

Fig. 9. Proposed architecture for the L&R algorithm

The block diagram of the proposed architecture is described in Fig. 9. The new architecture consists of 3 units.

First, Unit1 registers 36 pilot symbols from input symbols and passes the appropriate pilot symbol to Unit 2. A current pilot stands for a+jb and a delayed pilot does c+jd, respectively. After all of 36 pilots are stored in the shift registers, the overall estimator performs phase 0 to compute the real part of $R_n(k)$. In phase 0, the control logic selects the appropriate pilot symbol for the serial correlation through the multiplexers. To select the symbols, the control logic counts variable i and k which are defined in Eq. (5).

Second, Unit2 computes $R_n(k)$ and accumulates its result. During the estimation phase 0, the real part of $R_n(k)$ is computed. The front multiplexers select c and d to compute a x c and b x d by the *phase_sel* signal and the accumulator A computes the real part of $R_n(k)$. The accumulator A executes the first summation of (6). The division by L_p-k is realized by the next multiplier and multiplexer. Then, the second summation of (6) is executed by the accumulator B. The phase 0 requires 330 cycles (35+34+… +26+25). After the 330 cycles, the value of the accumulator B is locked in the upper register in Fig. 9.

For the estimation phase 1, the imaginary part of $R_n(k)$ is computed by the same method. At the end of the estimation phase 1, two locked estimation values are added by the accumulator C as the last summation of Eq.(6). Finally, in Unit3, the arctan unit converts the vector-form estimation value into the phase-form and the frequency estimator maintains the idle phase.

By the proposed architecture, the L&R frequency estimator has the same estimation value with the direct implementation while the hardware complexity is dramatically reduced.

IV. Implementation Results

The proposed frequency synchronizer has been modeled by Coware^{TM} SPW and implemented in Verilog HDL. Logic synthesis has been performed using Synplicity^{TM} Synplify Pro 7.7 and Xilinx^{TM} ISE 6.3i. The proposed architecture has been thoroughly verified onto the iPROVE^{TM} FPGA board having the Xilinx^{TM} Virtex II XC2V8000F1152. The maximum operating frequency is 33Mhz, which is capable of 25Msps baud rate. Table II shows the performance comparisons between the direct implementation and the proposed architecture.

TABLE II. Performance comparisons

Component	Direct implementation			Proposed architecture		
	LUTs	Num	Total	LUTs	Num	Total
20x20 Multiplier	893	66	58,938	893	3	2,679
20x20 Adder	59	66	3,894	59	5	295
20bit Register	2	46	92	2	78	156
Control logic	30	1	30	1,799	1	1,799
Total LUTs	**62,954 (100%)**			**4,929 (7.83%)**		

The direct implementation requires 66 multipliers, 66 adders, 46 registers and the dedicated control logic. In contrast, the proposed architecture requires only 3 multipliers, 5 adders, 78 registers and control logic. In the proposed architecture, the control logic requires 1,799 look-up-tables (LUTs) which is somewhat large compared with the direct implementation since the 20-bit 36-channel multiplexers and various control signals are utilized. Totally, the proposed architecture is realized as much as 7.83% of the direct implementation. The significant reduction of LUTs has been achieved by the proposed serial correlation architecture but the estimation performance maintains the same with the direct implementation.

V. Conclusions

This paper proposed the new efficient coarse frequency synchronizer for DVB-S2. The proposed coarse frequency synchronizer can reduce the frequency offset from 6.25% to below 0.03%. In addition, the proposed synchronizer reduced 95% multipliers and 92% adders compared with the original architecture of the L&R algorithm. As a result, the proposed synchronizer can saves the implementation cost and the power consumptions. The proposed synchronizer has been implemented with only 7.83% LUTs of the direct implementation.

Acknowledgment

This work was supported in part by the second stage of Brain Korea 21 Project in 2006, in part by the ITSOC program of MIC (Ministry of Information and Communication), and in part by IDEC (IC Design Education Center) and in part by the IT R&D program of MIC/IITA [2007-S008-01, Development of 21GHz Band Satellite Transmission Technology].

References

[1] Digital video broadcasting (DVB); Framing structure, channel coding and modulation for 11/12 GHz satellite services, EN300 421 (V1.1.2), European Telecommunications Standards Institute(ETSI), 1997.

[2] Digital video broadcasting (DVB); User guidelines for the second generation system for broadcasting, interactive services, news gathering and other broad-band satellite applications (DVB-S2), European Telecommunications Standards Institute (ETSI), TR 102 376, 2005.

[3] Feng-Wen Sun, Yimm Jiang, Lin-Nan Lee, "Frame synchronization and pilot structure for second generation DVB via satellite," in *Int. J. Satell. Commun. Network.*, vol. 22, no. 3, pp. 319-339, June 2004.

[4] A. Morello and V. Mignone, "DVB-S2: the second generation standard for satellite broad-band services," in *Proc. IEEE.*, vol. 94, no. 1, pp. 210-227, January 2006.

[5] U. Mengali and A. N. D'Andrea, *Synchronization Techniques for Digital Receivers*. New York: Plenum, 1997.

[6] J.W. Park, M.H. Sunwoo, P. S. Kim and D.I. Chang, "An efficient data-aided initial frequency synchornizer for DVB-S2," in *Proc. IEEE Workshop on Signal Processing Systems (SiPS)*, October 2007.

[7] M. Luise and R. Reggiannini, "Carrier frequency recovery in all-digital modems for burst-mode transmissions, " *IEEE Trans. Commun.*, vol. 43, no. 3, pp. 1169-1178, Mar. 1995.

[8] U. Mengali and M. Moreli, "Data-aided frequency estimation for burst digital transmission," *IEEE Trans. on Commun.*, vol. 45, no. 1, pp. 23-25, Jun. 1997.

[9] M. P. Fitz, "Planar filtered techniques for burst mode carrier synchronization," in *Proc. IEEE GLOBECOM΄91*, vol. 1, Dec. 1991, pp. 365-369.