TWO-STEP APRROACH FOR COARSE TIME SYNCHRONIZATION AND FREQUENCY OFFSET ESTIMATION FOR IEEE 802.16D SYSTEMS

Tae-Hwan Kim and In-Cheol Park

Department of Electrical Engineering
Korea Advanced Institute of Science and Technology (KAIST)
373-1 Guseong-dong, Yuseong-gu, Daejeon 305-701, Republic of Korea
{thkim, icpark}@ics.kaist.ac.kr

ABSTRACT

Targeting IEEE 802.16d systems, this paper presents a new approach for coarse time synchronization and carrier frequency offset estimation. In contrast to the previous architecture that usually computes both of them jointly within a unified auto-correlator, the proposed one performs them separately to achieve more reliable frequency synchronization and to reduce the overall hardware complexity by optimizing them individually. Experimental results show that the proposed architecture leads to better frequency synchronization compared to the previous joint estimation, and is more efficient in both respects of silicon area and power consumption.

Index Terms— Orthogonal Frequency Division Multiplexing, WiMAX, IEEE 802.16d, Coarse Time Synchronization, Carrier Frequency Offset Estimation

1. INTRODUCTION

IEEE 802.16d, known as fixed WiMAX, is a promising technology for wireless MAN due to its high data transmission rate and scalability. It is based on the orthogonal frequency division multiplexing (OFDM) that modulates the data distributed on a number of sub-channels by using orthogonal bases. The OFDM technique is robust to the frequency selectivity problem mainly caused by multi-path fading.

The purpose of the coarse time synchronization is to detect the preamble. Since this synchronization has a significant effect on the overall performance and power consumption of the receiver, it is one of the most important processing steps in the receiver. A failure in this synchronization leads to the missing or false alarming of a preamble. All of the subsequent processing steps in the receiver will be nothing if the synchronization fails. The power efficiency of this synchronization is also important in IEEE 802.16d systems, as the time to search for a preamble can be long [1]. During the search time, the synchronization block is active, whereas all the other blocks in the receiver can be deactivated to save energy. Thus the synchronization block should be turned on for quite a long time, making its power efficiency significantly important in achieving a low power receiver.

This paper deals with the coarse time synchronization and the carrier frequency offset (CFO) estimation, and proposes a new architecture for the coarse time synchronization in IEEE 802.16d systems. As the symbol timing offset is usually estimated more finely in the following processing, the resolution of this synchronization can be coarser [2], [3]. To reduce the power consumption and the area, the proposed architecture employs two decoupled data-paths each of which is optimized for its associated functionality. In addition to the architecture, a new synchronization schedule is also proposed to achieve more accurate synchronization.

The rest of this paper is organized as follows. In Section II, the previous architectures presented for the coarse time synchronization are analyzed. The proposed architecture is described in Section III. In Section IV, we evaluate the hardware complexity and the performance of the proposed architecture. Finally, concluding remarks are made in Section V.

2. PREVIOUS WORKS

Most of the previous synchronization schemes proposed for OFDM systems are based on the correlation between received samples and the training sequence. In this section, they are briefly explained along with their merits and drawbacks.

2.1. Auto-Correlation for Joint Estimation [4]

A data frame of IEEE 802.16d starts with a preamble like other burst-mode OFDM systems. As shown in Fig. 1, the downlink preamble consists of 4 short training sequences and 2 long training sequences [1]. The main objective of the repetitive short training sequences is to detect the preamble by performing auto-correlation between received samples [1]. To make the auto-correlation independent of the channel environment, the normalized

1-4244-1222-6/07/$25.00 ©2007 IEEE

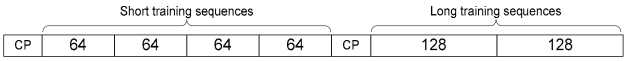


Fig. 1. Downlink preamble structure of IEEE 802.16d

auto-correlation defined below is usually employed for the synchronization [2-6],

$$\mu_n = \left(\sum_{i=1}^{W} r_{n-i} \cdot r^*_{n-i-W}\right) / \left(\sum_{i=1}^{W} |r_{n-i}|^2\right), \qquad (1)$$

where r is a received sample and W denotes the length of a short training sequence, which is 64 for the downlink preamble of IEEE 802.16d. The auto-correlation between the repetitive training sequences makes μ_n prominently high in the time domain, forming a magnitude plateau as shown in Fig. 2. In the receiver, the plateau is searched to detect the preamble. Searching for the plateau is usually achieved in practice by comparing the auto-correlation magnitude to a threshold. In addition, μ_n can also be used to estimate the CFO. If there is a CFO between the receiver and the transmitter frequencies, the phase difference is linearly increasing in the time domain. Therefore the phase difference between r_{n-i} and r_{n-i-W} is constant and proportional to W times the CFO if the CFO is assumed to be time-invariant. To estimate a more reliable CFO, we take the average of the phase differences of W received sample pairs as follows.

$$\varepsilon = \frac{1}{W} \angle \left(\sum_{i=1}^{W} r_n \cdot r^*_{n-i-W}\right), \qquad (2)$$

This joint estimation based on the auto-correlation of received samples is attractive because it is simple and robust. However, this scheme has some drawbacks to be improved. First, the previous architectures do not take into account the samples involved in the auto-correlation seriously. Most of them use a unified auto-correlator to obtain both of the time synchronization and the CFO estimation jointly at the same time [3], [5]. In other words, the auto-correlation used for the CFO estimation is also compared to a threshold in order to detect the plateau boundary for the coarse time synchronization. This can lead to a significant degradation of the CFO estimation. Let us assume that the auto-correlation magnitude becomes greater than the threshold as the received samples are serially fed into the auto-correlator. In this case, that point may be near the plateau but not exactly in the plateau. Examining the estimation window involved in the auto-correlation, we often find that some of the samples in the window are out of the training sequences, as depicted in Fig. 2. This can occur even if the timing is achieved by detecting the falling edge of the auto-correlation. Only in the plateau, we can guarantee that the estimation window corresponds to the training sequences.

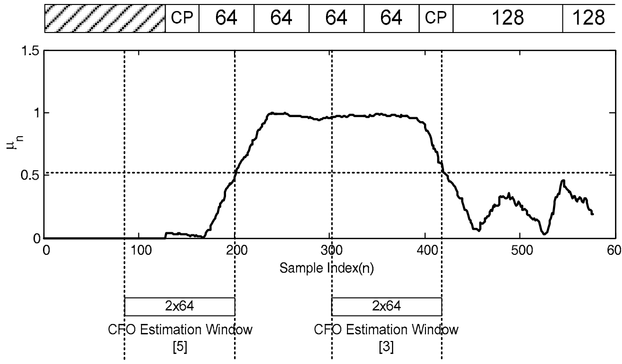


Fig. 2. Normalized auto-correlation of received samples for coarse time synchronization

Therefore, the CFO can be contaminated by some samples in the incorrect estimation window that are out of the same training sequences, if estimated jointly with the plateau detection. Some works have revised this problem by detecting a peak instead of the threshold comparison, but such an approach entails a considerably increased complexity [6]. Second, maintaining the same precision in the two estimations may be wasteful. In general, the time synchronization requires much less precision than the CFO estimation does. The coarse time synchronization can be successful even with reduced hardware exploiting this property [2].

2.2. Cross-Correlation for the Estimation of Symbol Timing Offset

In this scheme, the cross-correlation between received samples and pre-determined training sequences is performed to obtain the time synchronization by detecting the peak of the cross-correlation [7], [8]. The accuracy of this approach is fairly good even in a very low SNR. However, it can be applied only to the time synchronization, not to the CFO estimation. Moreover, the cross-correlation is very sensitive to the CFO, making the time synchronization unreliable if there is a small CFO. In general, this approach can be applied to CFO-corrected samples, and is appropriate for the fine symbol timing offset estimation to be performed after the received samples are corrected by compensating the CFO.

3. PROPOSED ARCHITECTURE

In contrast to the previous joint estimation, the proposed synchronization developed for IEEE 802.16d systems consists of two separate data-paths each of which is dedicated for its function.

3.1. Overall Synchronization Procedure

The overall flow of the proposed synchronization is depicted in Fig. 3, which works according to the following sequence.

i. Coarse time synchronization using auto-correlation
ii. CFO estimation using precise auto-correlation
iii. CFO correction
iv. Fine symbol timing

The symbol timing synchronization is achieved in two steps. The symbol timing offset (STO) is coarsely estimated by performing auto-correlation for the received samples, and then the fine STO estimation is performed for the CFO-corrected samples. In the coarse time synchronization, the received samples can be quantized to a few bits in order to reduce the computational complexity. The normalized auto-correlation performed for the quantized samples is compared to a threshold for this synchronization. The CFO is estimated by performing a precise auto-correlation only for *W* samples coming after the coarse time synchronization, and then is used to correct the received samples by using the CORDIC-like processing proposed in [10]. To determine the boundary of the short training sequences, the CFO corrected samples are involved in the fine STO estimation that performs cross-correlation for *2W* samples. Since the fine STO estimation follows the CFO correction step, this paper focuses on the first two steps, the coarse time synchronization and the CFO estimation.

Let x be the point detected by the coarse time synchronization. To work properly for the 4 short training sequences, x must satisfy the following relation when $W = 64$,

$$CP + 64 - t_a < x \leq CP + 64 \cdot 2 - t_a , \tag{5}$$

where t_a is the computational latency of the CFO estimation.

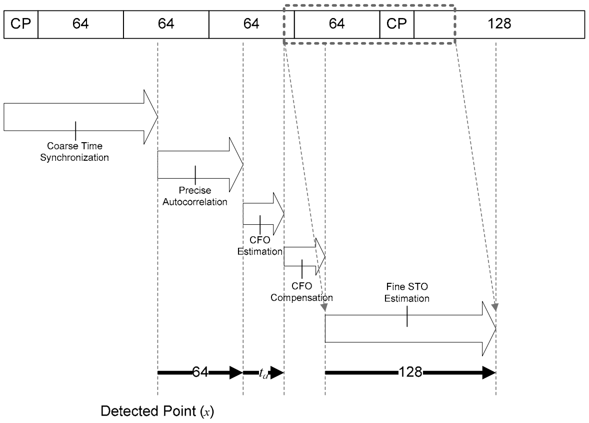


Fig. 3. Overall flow of the proposed synchronizations

The CFO can be estimated in no more than 16 clocks by using the architecture in [10].

3.2. Decoupled Data-Paths

In the previous architectures, the coarse time synchronization and the CFO estimation are jointly performed based on the unified auto-correlator [3], [5]. This approach is inefficient as the precisions required in the two auto-correlations differ from each other. Through a number of the fixed-point simulations the bit-widths can be determined. The bit-width of the CFO estimation is about 10 bits to obtain a reasonable performance, while only 4 bits is sufficient for the coarse time synchronization. According to this observation, the proposed architecture has two separate data-paths to optimize each auto-correlation individually, as shown in Fig. 4. The upper part is for the coarse time synchronization expressed in (6), and the lower part is for the precise auto-correlation required for the CFO estimation defined in (7).

$$\mu_n = \frac{\left| \sum_{i=1}^{W} r_{n-i} \cdot r_{n-i-W}^{*} \right|^2}{\left| \sum_{i=1}^{W} | r_{n-i} |^2 \right|^2} \tag{6}$$

$$\Lambda_n = \sum_{i=1}^{W} r_{n-i} \cdot r_{n-i-W}^{*} \tag{7}$$

Both of them calculate auto-correlation, but their bit-widths are different. In other words, r_n in (6) is represented in a smaller bit-width than r_n in (7) is.

3.3. Auto-Correlator for Coarse Time Synchronization

To derive a more efficient computational structure, the auto-correlation is transformed to a recursive form [4] as follows,

$$\Lambda_n = \sum_{i=1}^{W} r_{n-i} \cdot r_{n-i-W}^{*} = \Lambda_{n-1} + r_{n-1} \cdot r_{n-W}^{*} - r_{n-W} \cdot r_{n-2W+1}^{*}$$
$$. \tag{8}$$

The recursive auto-correlation requires $2W$ sample buffers, which is equal to 128, for the short training sequences of the downlink preamble. This is implemented as a circular buffer in the proposed architecture, as shown in Fig. 4. In the coarse time synchronization, μ_n is compared

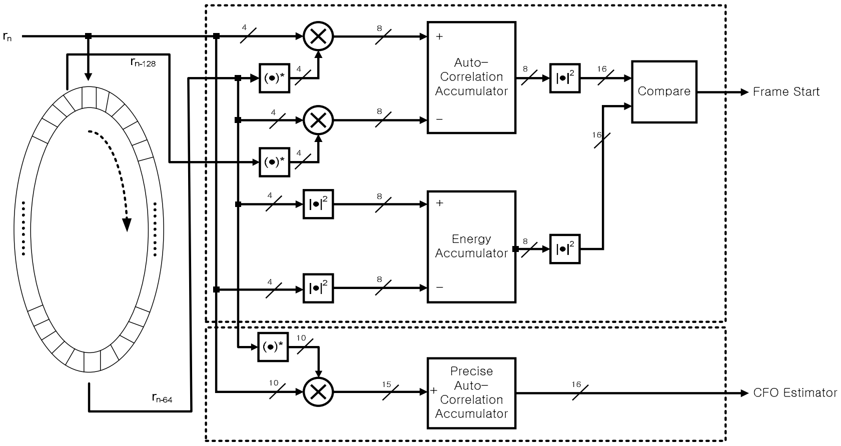


Fig. 4. Proposed architecture for the coarse time synchronization and precise auto correlation used for the CFO estimation

to a threshold, which can be efficiently implemented with no division as follows,

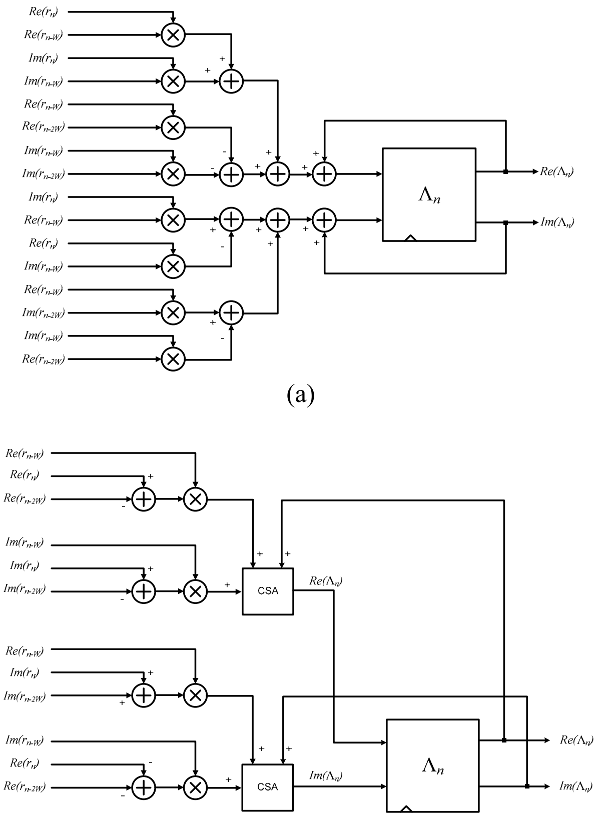
$$\left|\sum_{i=1}^{W} r_{n-i}\cdot r_{n-i-W}^{*}\right|^{2} - Th\cdot\left|\sum_{i=1}^{W} |\, r_{n-i}\,|^{2}\right|^{2} > 0 \quad , \qquad (9)$$

where *Th* means the threshold. If the threshold is a power of two, the multiplication of the second term can be replaced with a shift operation. Through a number of simulations, the threshold is determined to 1/4. As stated before, the coarse time synchronization can be successful even if the samples are represented in a small bit-width. Reducing the bit-width helps lower the power consumed in the synchronization.

The recursive auto-correlation expressed in (8) may be implemented in a straightforward structure, as depicted in Fig. 5(a). In this figure, we can find that a pair of the two multiplications has a common operand such as $Re(r_{n-W})$, $Im(r_{n-W})$, $Re(r_{n-W})$ or $Im(r_{n-W})$. The other operands can be added before the multiplication to reduce the number of multipliers, which enables eight n x n multiplications to be implemented with four n x $(n+1)$ multipliers as shown in Fig. 5(b). Additionally, the carry save addition is employed to add three operands in a more economical way.

3.4. Auto-correlator for CFO estimation

A separate data-path is used for the auto-correlation of the CFO estimation, as it needs a relatively high precision. In this case, the auto-correlation does not need to be normalized, since only the average phase difference is needed in the CFO estimation. Therefore, the multipliers and the accumulator needed for calculating energy can be eliminated as illustrated in Fig. 4. Moreover, the auto-correlation is calculated only once for W samples just after the coarse time synchronization is completed as shown in Fig. 3. It can be calculated with a complex multiplier and an accumulator more efficiently than the recursive form that needs two complex multipliers, as shown in Fig. 6. Though the recursive form is effective in computing many auto-correlations for a sequence of samples, it is not needed in



(a)

(b)

Fig. 5. Auto-correlator for the coarse time synchronization (a) Straight-forward structure (b) Proposed structure

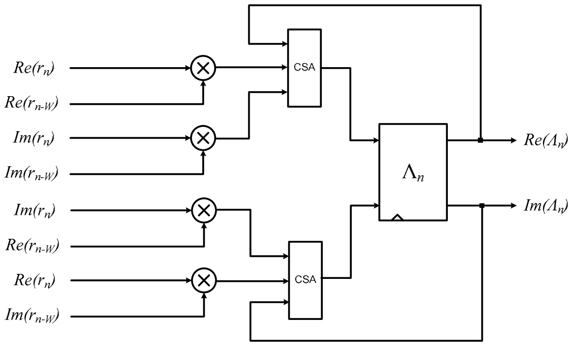


Fig. 6. Auto-correlator for the CFO estimation

the proposed architecture, as the CFO estimation deals with only W samples coming after the coarse time synchronization.

In addition, the separate auto-correlation has an advantage that the CFO estimation is more accurate than the previous joint estimation. As stated before, the auto-correlation of the CFO estimation should be more precise than that of the coarse time synchronization. In the proposed architecture, W samples in the CFO estimation window coming after the coarse time synchronization are considered, whereas the previous architectures usually estimate the CFO based on the auto-correlation calculated at the moment when the preamble is detected by comparing μ_n to a threshold [3], [5]. This can cause a negative effect on the estimation performance because some incorrect samples can be included in the estimation window as we explained in the previous section. As shown in Fig. 7, however, W samples in the estimation window of the proposed architecture are guaranteed to be within the short training sequences, because they are taken after the coarse time synchronization is done. Therefore we can expect that the accuracy of CFO estimation is significantly improved due to the purity of the samples involved in this precise auto-correlation.

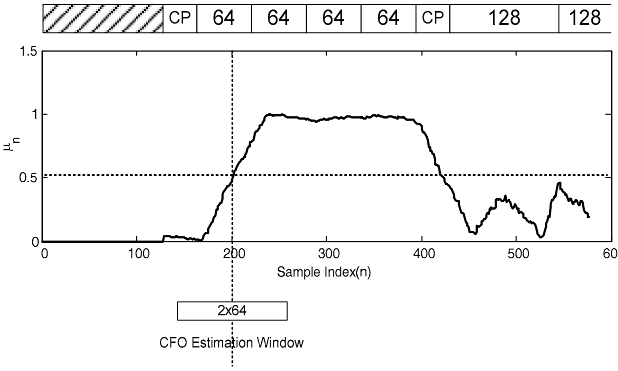


Fig. 7. CFO estimation window in the proposed synchronization

4. EVALUATIONS

The performance of the proposed architecture is evaluated through simulations performed with a multi-path fading channel whose parameters are conformed to SUI-3 for all the SNR range specified in IEEE 802.16d [1], [9]. The CFO in the channel is set to 1.7 sub-carrier spacings in the simulations.

The CFO estimation errors are compared in Fig. 8, where the proposed architecture performs auto-correlation for the W samples coming after the coarse time synchronization and the previous approach shares the auto-correlation used for the coarse time synchronization. As there is no ambiguity in selecting samples to be involved in this estimation, the proposed architecture shows better performance than the previous one.

Fig. 9 compares the performance of the coarse time synchronization. If the detected point violates the relation described in the previous section, it is regarded as a failure. Samples are quantized to 4 bits in the proposed architecture, while they are represented in 10 bits in the previous unified architecture. We can see in the figure that the proposed architecture performs comparably to the previous architecture, even though the samples are represented in a much less bit-width.

The previous joint estimation and the proposed architecture are implemented with TSMC 0.25um CMOS standard cell library and synthesized under the same constraints. The gate counts resulting from the two approaches are compared in Table I. The gate counts are normalized by counting a 2-input NAND cell as one. As shown in the Table I, the proposed architecture achieves approximately 30% gate count reduction in spite of the two separate data-paths. Compared to the joint architecture, the proposed architecture has more multipliers, but significantly reduces the bit-widths of some multipliers. As the complexity of a multiplier is squarely proportional to its bit-width, reducing bit-width pays for the increased number of multipliers in the proposed architecture.

To investigate the power efficiency of the proposed architecture, we simulated the power consumption for the gate-level net-lists. The simulations were performed with the test vectors generated for the multi-path fading channels described above. The results are summarized in Table II. Looking at the results, we can see that the proposed architecture reduces the power consumption to about 40% compared to the previous one. This is a natural consequence, since each data-path is individually optimized in the proposed architecture and the high-precision data-path associated with the CFO estimation is active only for a tiny interval. Other blocks not involved in the synchronization are turned off to save power consumption during this synchronization. In IEEE 802.16d, the synchronization time can be as long as 2 MAC frames [1]. This means that the power efficiency of the synchronization block has a significant effect on the overall power consumption of the target system.

5. CONCLUSION

In this paper, we have proposed a new synchronization architecture that is good in performance and efficient in both respects of power and area consumption. In contrast to the joint architecture with a unified auto-correlator, the proposed one performs them separately to achieve more reliable frequency synchronization and to reduce the overall hardware complexity by optimizing them individually. The coarse time synchronization can be successfully computed with a small bit-width, leading to a simplified hardware. In addition, the CFO estimation can be made accurate by eliminating the incorrect samples that do not correspond to the training sequences. Compared to the previous joint estimation approach, the proposed synchronization block implemented for IEEE 802.16d systems reduces silicon area and power consumption by 30 % and 60 %, respectively.

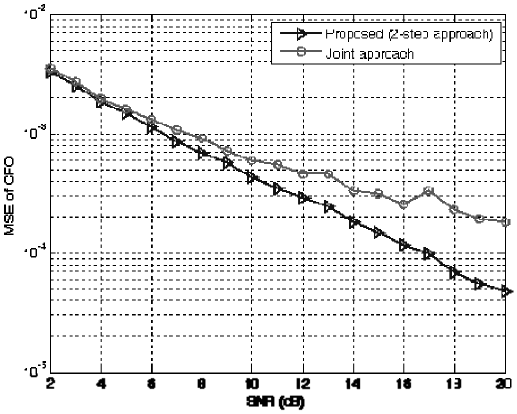


Fig. 8. Performance result of CFO estimation

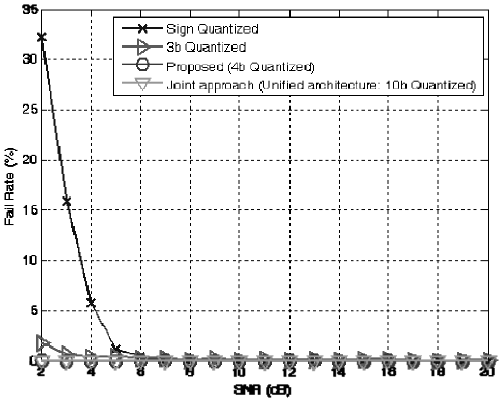


Fig. 9. Performance result of coarse time synchronization

TABLE I. GATE COUNT COMPARISON

	Proposed	*Joint Approach*
Precise Auto-Correlator	4232	11321
Coarse Time Synch.	3688	
Controller	165	165
Buffer Control	319	319
Etc.	270	266
Total.	8674	12071

TABLE II. POWER CONSUMPTION COMPARISON

SNR (*dB*)	*Proposed* (*mW*)	*Joint Approach* (*mW*)
2	6.195	14.630
5	6.087	15.306
8	6.123	16.331
11	5.924	15.127
14	5.876	15.680
17	5.777	16.187
20	5.769	15.917

ACKNOWLEDGEMENT

This work was supported by Institute of Information Technology Assessment through the ITRC.

REFERENCES

[1] IEEE Std. 802.16. IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Fixed Broadband Wireless Access Systems, 2004.

[2] M. Speth, S. A. Fechtel, G. Fock, and H. Meyr, "Optimum Receiver Design for OFDM-Based Broadband Transmission - Part II: A Case Study," IEEE Transactions on Communications, vol. 49, no. 4, April 2001.

[3] Mody, A.N. and Stűber, G.L., “Receiver implementation for a MIMO OFDM system,” in Proc. of Global Telecommunications Conference 2002, vol.1, pp716-720.

[4] Schmidl. T.M. and Cox, D.C, “Robust frequency and timing synchronization for OFDM,” IEEE Trans. on Communications, pp1613-1621, Dec 1997

[5] Fort.A. and Eberle.W., “Synchronization and AGC proposal for IEEE 802.11a burst OFDM systems,” Global Telecommunications Conference 2003, vol.3, pp1335-1338

[6] Ch. Nanda Kishore and V. Umapathi Reddy, “A frame synchronization and frequency offset estimation algorithm for OFDM system and its analysis,” EURASIP Journal on Wireless Communications and Networking, vol. 2006, Article ID 57018, 16 pages, 2006

[7] Tufvesson F. and Edfors O. et al., “Time and frequency synchronization for OFDM using PN-sequence preambles,” Vehicular Technology Conference 1999, pp2203-2207, vol.4

[8] Kun-Wah Yip and Yik-Chung Wu et al., “Design of multiplierless correlators for timing synchronization in IEEE 802.11a Wireless LANs,” IEEE Trans. on Consumer Electronics, pp107-114, Feb. 2003

[9] V.Erceg, K.V.S Hari, M.S. Smith et al., “Channel models for fixed wireless applications,” Contribution IEEE 802.16.3c-01/29r1, Feb. 2001

[10] Maharatna K., Baneriee S. et al., “Modified virtually scaling-free adaptive CORDIC rotator algorithm and architecture,” IEEE Trans. on Video Technology, pp1463-1474, Nov. 2005