

# A Block-Parallel Architecture for Initial and Fine Synchronization in OFDM Systems

Pramod Udupa, Olivier Sentieys, and Pascal Scalart  
 IRISA, University of Rennes 1, France  
 Email: {pudupa, sentieys, scalart}@irisa.fr

**Abstract**—A novel low complexity parallel algorithm and its associated architecture are proposed for initial synchronization in orthogonal frequency division multiplexing (OFDM) systems. The method is hierarchical and uses auto-correlation for the first step and cross-correlation for the second step. The main advantage of the proposed approach is that it reduces the computational complexity by a factor of five (80%), while achieving similar mean square error (MSE) as cross-correlation based methods. The method uses block-level parallelism for auto-correlation step, which speeds up the computation significantly. After fixed-point analysis, a parallel architecture is proposed to accelerate both coarse and fine synchronization steps. This parallel architecture is scalable and provides speed-up proportional to number of parallel blocks.

## I. INTRODUCTION

OFDM systems are very sensitive to timing and frequency synchronization [1] errors. Loss of timing synchronization causes inter-carrier interference (ICI) and inter-symbol interference (ISI). Also, frequency offset causes loss of orthogonality among sub-carriers which leads to ICI. These synchronization errors, if left uncompensated, reduces the advantage provided by OFDM systems over single-carrier systems. The initial synchronization step is also very important for receiver performance.

Initial timing synchronization can be classified based on whether it is done at the uplink [1] [2] or downlink. Uplink synchronization is done at the base station (BS) where it synchronizes symbols received from multiple users. BS synchronization deals with signals received from different users, where each signal propagates through different multipath channel and complicates the problem. Downlink synchronization occurs at the mobile station (MS) where it processes signal from BS. Here we consider the problem of downlink synchronization using preamble symbols.

Initial timing synchronization methods for OFDM are either based on preamble symbols [3] [4] [5] or cyclic prefix (CP). Preamble-based synchronization methods use specially designed training sequence (TS) to support methods based on auto-correlation [3], cross-correlation [4], or both [5]. The auto-correlation method uses multiple repeating parts of a training symbol (TS) with different sign patterns like  $[+1 -1 +1 +1]$  to maximize the peak and reduce the side lobes of the timing metric. The cross-correlation method uses either cross-correlation with transmitted symbol or with itself for detection. A central-symmetric sequence can be used for cross-correlation operation like in  $[A B^*]$ ,  $B = A[-n]$ . The

main disadvantage of cross-correlation methods is that they require lot of computation resources for a reasonably long sequence (e.g.  $N_p = 64$ ).

A third scheme which combines both auto-correlation and cross-correlation is called hybrid detection scheme [6]. This scheme uses training symbol which has repeated parts and is centrally symmetric. The method proposed by Zhou [5] uses the multiplication of both correlation metric in the same equation to achieve better performance. The method uses multiple thresholds to detect the first path in a frequency selective fading channel. However, the overall computation time is significantly higher due to cross-correlation computed together with auto-correlation and this can lead to low throughput and large delay in establishing synchronization.

In this paper, a new hierarchical estimation scheme is proposed. It provides performance close to or better than cross-correlation but at a significantly lower complexity. The proposed method uses auto-correlation to identify the start window of training symbol and then uses cross-correlation with known TS to identify the first arrival path. The threshold for fine synchronization is found using noise statistics. A block-level parallel architecture is proposed which speeds up the computation. The block-level parallel algorithm is implemented on a FPGA for throughput evaluation. An accuracy analysis of the algorithm using fixed-point arithmetic is also performed to evaluate the performance degradation due to lower precision.

The paper is arranged as follows. In Section II, the system model used for the timing synchronization method is described. Section III explains the new timing synchronization method, while Section IV gives the simulation results for performance comparison with previous approaches. Section V describes algorithm complexity and the proposed parallel architecture and details fixed point analysis and implementation results. Finally, Section VI concludes the paper.

## II. OFDM SIGNAL DESCRIPTION

The transmitted baseband OFDM samples can be written in terms of IFFT equation as

$$x[n] = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} X[k] \cdot e^{j2\pi nk/N} \quad (1)$$

where  $N$  is the number of sub-carriers and  $X[k]$  the complex information carrying symbol in frequency domain. The

sampled signal at the receiver can be written as

$$r[n] = s[n - \eta] \cdot e^{j(2\pi\epsilon n/N + \phi)} + w[n] \quad (2)$$

where  $\eta$  is the integer timing offset,  $\epsilon$  the frequency offset and  $\phi$  the phase offset.  $w[n]$  is the additive white Gaussian noise (AWGN) and  $s[n]$  in multipath channel is given by

$$s[n] = \sum_{m=0}^{L_h-1} h[m] \cdot x[n - \tau_m] \quad (3)$$

where  $h$  is the sampled channel response (complex channel coefficients) at the receiver.  $L_h$  is the number of channel paths and  $\tau_m$  is the path delay corresponding to the  $m^{\text{th}}$  channel path. The channel is assumed static for the duration of the OFDM symbol.

### III. PROPOSED TIMING SYNCHRONIZATION

Timing synchronization is the step of finding the starting of the OFDM symbol. Frame and symbol synchronization follows directly from timing synchronization in OFDM. Timing synchronization involves finding the unknown time offset ( $\eta$ ) in the presence of other unknowns ( $\epsilon$  and  $\phi$ ).

The training sequence proposed is based on Zadoff-Chu sequence, which has low PAPR and ideal auto-correlation properties. The sequence is given by

$$a_k^{(r)} = \begin{cases} \exp\left(i \frac{2\pi}{N_s} \left\lfloor \frac{rk^2}{2} \right\rfloor\right), & \text{for } N_s \text{ even} \\ \exp\left(i \frac{2r\pi k(k+1)}{N_s}\right), & \text{for } N_s \text{ odd} \end{cases} \quad (4)$$

where  $0 \leq k < N_s$ ,  $\text{gcd}(r, N_s) = 1$ . This sequence is mapped to every fourth subcarrier of sequence of length  $N = 512$ . After IFFT and multiplying by the pattern  $[+1 +1 +1 -1]$ , the proposed training symbol is generated. The proposed training symbol (TS) is

$$[A \ A \ A \ -A]$$

The sign pattern  $[+1 +1 +1 -1]$  is designed to ensure steep roll-off for initial estimation algorithm. The proposed synchronization method is a hybrid method using auto-correlation in the first step and then cross-correlation with the known transmitted sequence.

#### A. Auto Correlation

The auto-correlation step uses all the four repeating parts of the TS to achieve steep timing metric roll-off. The timing metric for coarse initial estimate is:

$$M_{init}[n] = \frac{|P[n]|^2}{R^2[n]} \quad (6)$$

where  $P[n]$  is the auto-correlation function,  $R[n]$  is the energy of the sequence and  $L$  is the number of repeating parts ( $L = 4$ )

in the training symbol. The expressions for  $P[n]$  and  $R[n]$  are

$$P[n] = \sum_{k=0}^{L-2} u[k] \sum_{m=0}^{M-1} r^*[n + kM + m] \cdot r[n + (k+1)M + m] \quad (7a)$$

$$R[n] = \sum_{k=1}^{L-1} \sum_{m=0}^{M-1} |r[n + kM + m]|^2 \quad (7b)$$

where  $u[k] = p[k] \cdot p[k+1]$ ,  $p[k]$  contains the sign pattern of  $[+1 +1 +1 -1]$  with  $k = 0, 1, \dots, (L-1)$  and  $M = N/L$ . The time index corresponding to the maximum value gives the initial estimate.

$$\hat{\eta}_{init} = \arg \max_n M_{init}[n] \quad (8)$$

Figure 1a shows the plot of  $M_{init}[n]$  for a Signal-to-Noise Ratio (SNR) of 6 dB in a frequency selective channel. The maximum peak is not exactly at zero which is the actual start of the OFDM symbol, but slightly shifted to the right due to multipath effect. The fine estimation algorithm consists of correcting this unknown shift and finding the correct starting point. The fine estimation algorithm does not assume a dominant first path and can work with non-dominant first path in multipath channel.

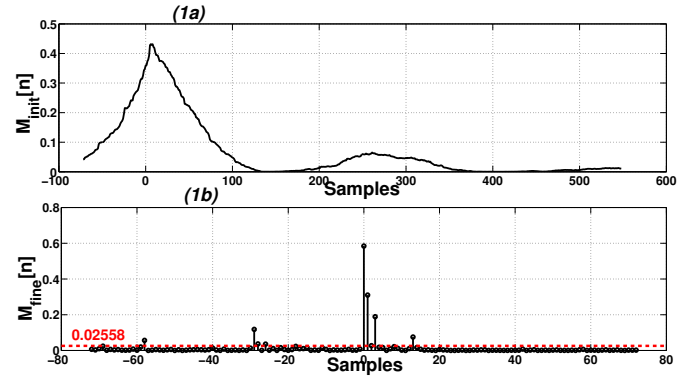


Fig. 1. Plot of Coarse (a) and Fine (b) Timing Metric Functions

#### B. Cross Correlation

The fine estimation algorithm uses cross-correlation with part of the repeated transmitted symbol  $A$ . Using the calculated  $\eta_{init}$  value, a search window of  $[-N_{cyp}, N_{cyp}]$  is used for the cross-correlation process. The conjugate of  $A$  is given by  $y[n] = A^*[n]$ . The length of  $y[n]$  is  $M = \frac{N}{4}$ . The timing metric is then given by

$$M_{fine}[n] = \frac{\sum_{k=0}^{\frac{N}{4}-1} r[n+k] \cdot y[n+k]}{\sum_{k=0}^{\frac{N}{4}-1} |r[n+k]|^2} \quad (9)$$

where  $r[n]$  is the received sequence. This range for  $n$  was chosen since the initial estimate does not produce peaks outside the length of cyclic prefix ( $N_{cyp}$ ). Figure 1b shows the

plot of  $M_{fine}[n]$  for a SNR of 6 dB. The time index of the maximum value of  $M_{fine}[n]$

$$\hat{\eta}_{fine} = \arg \max_n (M_{fine}[n]) \quad (10)$$

corresponds to start of the TS. In case of channel with large delay,  $\eta_{fine}$  may not correspond to the first arrival path. To find the first path in a multipath channel, a threshold based summation method is proposed.

### C. Threshold Based Summation

The values of  $M_{fine}[n]$  are thresholded by a value  $\beta$  such as

$$M_{fine}[n] = \begin{cases} M_{fine}[n], & M_{fine}[n] > \beta, \\ 0, & \text{otherwise,} \end{cases} \quad (11)$$

$\beta$  is the threshold which separates signal and noise components in  $M_{fine}[n]$  and is determined using the probability distribution of the noise component in  $M_{fine}[n]$ . The steps are as follows:

- The sequence  $M_{fine}[n]$  is passed through Lloyd-Max [7] quantization algorithm using three levels of quantization.
- The lowest quantization level and its cluster is considered as noise here. It is observed that this follows a lognormal distribution. Mean ( $\mu_n$ ) and variance ( $\sigma_n^2$ ) of noise cluster are calculated first. The corresponding mean  $\mu$  and  $\sigma$  for lognormal distribution is

$$\mu = \log \left( \frac{\mu_n^2}{\sqrt{\sigma_n^2 + \mu_n^2}} \right) \quad (12a)$$

$$\sigma = \sqrt{\log \left( \frac{\sigma_n^2}{\mu_n^2} + 1 \right)} \quad (12b)$$

- A constant false alarm rate (CFAR) of  $\alpha$  is used for calculation of threshold. The equation for threshold is derived by integrating probability distribution function (pdf) with limits  $[\beta, \infty]$ , which leads to

$$\beta = e^{(\sqrt{2} \cdot \sigma \cdot \text{erf}^{-1}(1-2\alpha) + \mu)} \quad (13)$$

For a fast-varying channel, the threshold  $\beta$  needs to be recalculated regularly. But on a slow-varying channel,  $\beta$  could be calculated offline and stored in a look-up table (LUT). A CFAR is maintained across all SNR values. A windowed summation is performed after discarding the noise values using the threshold calculated.

$$E_p(n) = \sum_{k=0}^{S_w-1} M_{fine}(\hat{\eta}_{fine} - n + k) \quad (14)$$

where  $S_w$  is the length of summation window (root mean square (RMS) delay of channel) and  $J_m$  is the search window (maximum delay of channel). The index of the first arrival path is calculated using windowed summation, which is given by

$$\hat{\eta}_{first} = \arg \max_n E_p(n) : n = 0, 1, \dots, J_m \quad (15)$$

Finally,

$$\hat{\eta}_{final} = \hat{\eta}_{init} - \hat{\eta}_{first} \quad (16)$$

indicates final estimate of the starting index of the OFDM symbol.

CFO can be calculated from  $P[\eta_{final}]$  using the formula.

$$\hat{\epsilon} = \frac{2 * \angle(P[\eta_{final}])}{\pi} \quad (17)$$

The frequency offset estimation range of  $\hat{\epsilon}$  for proposed TS is  $\pm 2$  sub-carrier spacing.

## IV. SIMULATION RESULTS

### A. Parameters

The performance of all synchronization algorithms has been investigated by using Monte-Carlo simulations. Algorithms using both auto-correlation, cross-correlation and hybrid techniques are compared in this section. The OFDM system parameters are shown in Table I. The channel used here is a frequency selective channel (ISI channel) with an exponential Power Delay Profile (PDP) and ratio of first to last Rayleigh fading tap is set to 20 dB. The channel has 15 taps with equal tap spacing of four samples.

TABLE I  
SIMULATION PARAMETERS

Number of Tx Antennas	2
Number of Rx Antennas	2
Sampling Frequency(MHz)( $F_s$ )	15.36
IFFT/FFT Size ( $N$ )	1024
Length of Cyclic Prefix ( $N_{cyp}$ )	72
Window Size $S_w$ (samples)	40
Distance $J_m$ (samples)	36
Constant False alarm rate ( $\alpha$ )	0.01
Number of simulation runs	$10^4$
Number of channel taps	15
Ratio of first to last tap (in dB)	20
Carrier Frequency Offset (CFO)( $\epsilon$ )	0.75

### B. MSE of Timing Estimation

The delay correlation based method (Minn) [3] has higher MSE compared to methods using cross-correlation (Choi) [4] only. The performance of the proposed method (*Proposed(final)*) is better compared to Minn, and comparable to Choi at significantly lower computational complexity as shown by Figure 2. *Proposed(coarse)* in Figure 2 shows the performance of coarse synchronization algorithm.

### C. MSE of Frequency Estimation

Figure 3 shows the MSE of frequency estimation over a range of SNR values. A CFO of 0.75 was used during these simulations. Choi's method is not represented since no frequency estimation is proposed it. The Cramér-Rao bound for variance in estimation of frequency offset [8] is given by

$$CRB(\hat{\epsilon}) = \frac{1}{2\pi^2} \frac{3(SNR)^{-1}}{N(1-1/N^2)} \quad (18)$$

where  $N$  is the size of FFT. Minn's estimator uses the algorithm of Morelli [8], which is computationally more complex compared to proposed method.

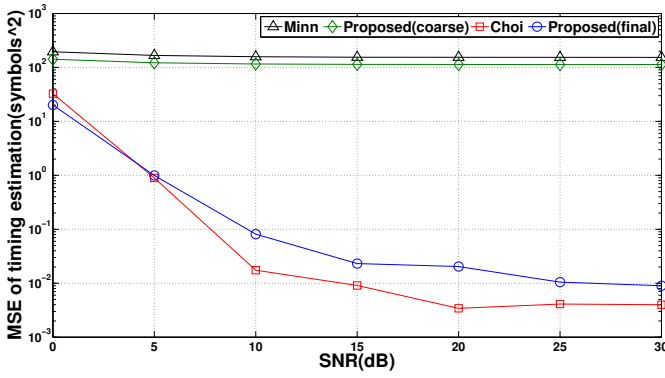


Fig. 2. MSE of Timing Estimation versus SNR

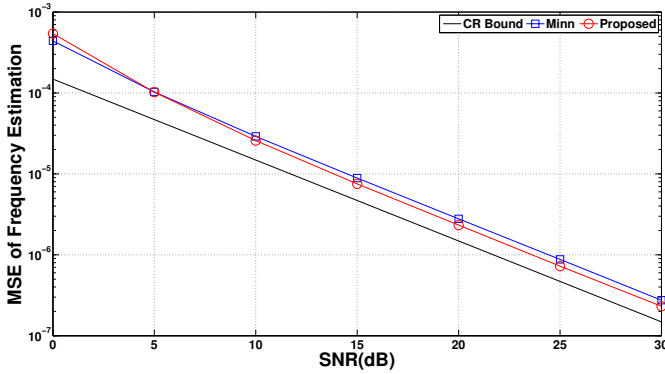


Fig. 3. MSE of Frequency Estimation versus SNR

## V. PROPOSED PARALLEL ARCHITECTURE

### A. Complexity Analysis

The number of real operations required per symbol is shown in Table II, where  $N_{sym} = N + N_{cyp}$  is the total OFDM symbol length. Since the algorithms of Minn can be written in an iterative form, the number of operations is fixed per point after the initial calculation. But for Choi and Zhou [5] algorithm, where the numerator part cannot be written in an iterative form, the number of operations becomes a function of the FFT size. Since the proposed algorithm has initial iterative part and fine non-iterative part, it is also a function of  $N$ . The non-iterative part only operates on length of  $(2N_{cyp} + 1)$  which is very much lesser than  $N_{sym}$ .

TABLE II  
NUMBER OF OPERATIONS PER OFDM SYMBOL

Algorithm	Real Multiplication	Real Addition	Division
Minn	$31N_{sym}$	$29N_{sym}$	$N_{sym}$
Choi	$(2N + 7)N_{sym}$	$(2N + 3)N_{sym}$	$N_{sym}$
Zhou	$(2N + 11)N_{sym}$	$(2N + 11)N_{sym}$	$N_{sym}$
Proposed coarse step	$31N_{sym}$	$29N_{sym}$	$N_{sym}$
Proposed fine step	$N$ $(2N_{cyp} + 1)$	$(N - 2)$ $(2N_{cyp} + 1)$	$1$ $(2N_{cyp} + 1)$

Comparing the proposed approach with Choi's and Zhou's

method, a saving of  $\frac{N_{sym} - (2 \cdot N_{cyp} + 1)}{N_{sym}}$  is obtained for all three basic operations. This results in approximately 80% reduction in computational complexity which means that the proposed method is 5 times less complex than other approaches in the literature with equivalent performance for timing synchronization and frequency estimation.

### B. Parallel Architecture

A block-level parallel algorithm and architecture is proposed in this section. The algorithm is transformed to achieve parallelism in both initial and fine search method. The equations of auto correlation can be written in an iterative form.

$$P[n+1] = P[n] - r^*[n] \cdot r[n+M] + 2r^*[n+2M] \cdot r[n+3M] - r^*[n+3M] \cdot r[n+4M] \quad (19)$$

$$R[n+1] = R[n] + |r[n+4M]|^2 - |r[n+M]|^2 \quad (20)$$

Although these equations provide low complexity update, their parallelization requires a large number of adders and dependency graph cannot be easily broken. To achieve a large amount of parallelism without significant increase in resources, the idea is to use block-level parallelism, in which multiple blocks work together simultaneously to compute outputs every cycle. Block-level parallelism is realized by computing specific intermediate points like  $0, M, 2M$ . using non-iterative formula (Equation 7a) and subsequent  $M - 1$  points using iterative formula of Equations 19 and 20. In each block, the initial point takes  $M = \frac{N}{4}$  cycles, while subsequent iterative updates take a single cycle for update. Multiple blocks operate on blocks of data and thus implement a parallel search procedure. The proposed architecture for 4-Parallel blocks is shown in Figure 4, where the first block is computing  $M_{init}$  for  $M$  output indices and the second block computes for next  $M$  indices. The increase in parallel memory is linear with P-parallel blocks to (P+1)-parallel operating blocks.

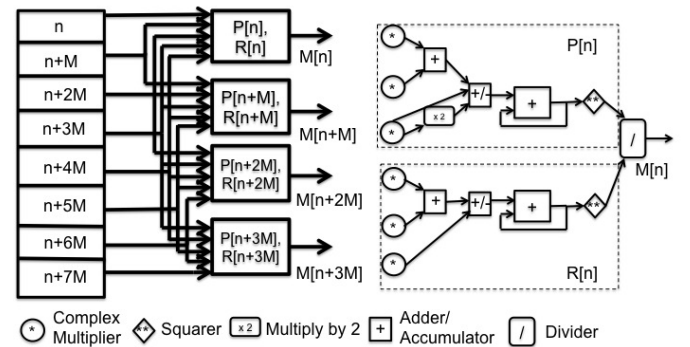


Fig. 4. 4-Parallel Architecture for Coarse Synchronization Algorithm

Figure 5 shows the mapping for fine synchronization algorithm. The amount of parallelism achieved here is three blocks in parallel compared to four of coarse synchronization. The total search space of  $2N_{cyp}$  is divided into three regions and searched by three parallel blocks. Computation is done using

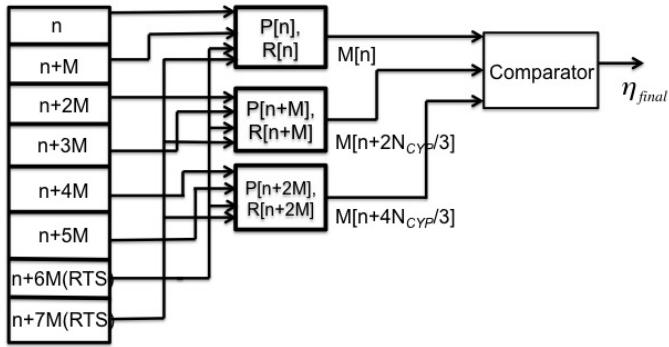


Fig. 5. 4-Parallel Architecture for Fine Synchronization Algorithm. Reference Training Symbol (RTS),  $N_{cyp}$  - cyclic prefix length

non-iterative form of Equation 9. Speed up is achieved using same architecture for both coarse and fine synchronization algorithms and is scalable to large number of parallel blocks.

### C. Fixed Point Analysis

Fixed point analysis of the proposed architecture is done to determine the minimum input precision at which the computation can be done. Using MSE of frequency estimation as a criterion, minimum acceptable precision is decided. Figure 6 shows the performance of the algorithm at various input precision widths for different SNR.

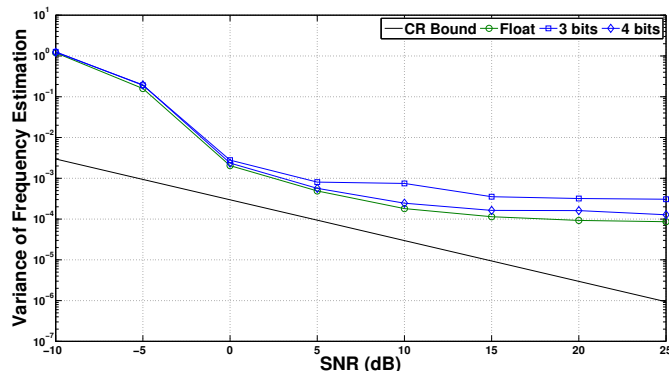


Fig. 6. MSE of Frequency Estimation versus SNR, # bits indicate represent fractional bits only, one extra bit is used for sign, so 3 bits + 1 sign bit = 4 bits

It is found that a precision of 5 bits is sufficient for the coarse grained synchronization step. The fine grained synchronization step can operate at very low precision like a single bit [5]. Lower the precision, higher is the gain in area obtained when more parallel blocks are added.

### D. Implementation Results

The proposed architecture has been implemented using CatapultC high level synthesis (HLS) tool on a Xilinx Virtex-6 FPGA with operating frequency of 150MHz, at input precision of 5 bits. Table III shows the throughput for  $M_{init}$  as function of number of parallel blocks. The third and fourth columns show the area occupied in the FPGA. Divider for calculation of  $M_{init}$  and  $M_{fine}$  is implemented using LUT.

TABLE III  
THROUGHPUT OF PARALLEL ARCHITECTURE WITH DIFFERENT BLOCKS

Parallel Blocks	Throughput Cycles	Area Score	Slice LUTs	LUT-FF Pairs	Parallelism C(F)
1(Sequential)	13286	12058	2544	2641	1(1)
2(Parallel)	7048	24638	4413	4544	2(2)
4(Parallel)	4082	35318	8067	8246	4(3)
8(Parallel)	2345	49611	9836	9992	8(5)

The throughput cycles reduces by nearly half for doubling the number of parallel blocks, i.e going from 2 to 4. Area score given by Catapult is number of LUTs. The last column indicates parallelism achieved individually for coarse and fine synchronization algorithm. The number inside bracket is for fine synchronization algorithm.

### VI. CONCLUSION

A novel hierarchical low complexity timing estimation algorithm is proposed which reduces the computational complexity by a factor of five (80%) to achieve the same MSE performance as Choi's [4] method. A novel block-level parallel algorithm is proposed for the auto-correlation operation, which can provide significant acceleration proportional to number of parallel blocks present. Same architecture can be reused for accelerating fine synchronization algorithm. Due to simpler structure of each block, the area increase is not too significant compared to the throughput improvement. Based on area constraints, number of parallel blocks can be decided. Thus, the proposed algorithm provides a very good trade-off between accuracy in estimation and computational efficiency, while block-level parallel algorithm gives scalable throughput improvement.

### ACKNOWLEDGEMENT

This work was partly supported by the 100GFLEX FUI project.

### REFERENCES

- [1] M. Morelli, C.-C. Kuo, and M.-O. Pun, "Synchronization techniques for orthogonal frequency division multiple access (OFDMA): A tutorial review," *Proceedings of the IEEE*, vol. 95, no. 7, pp. 1394–1427, July 2007.
- [2] L. Sanguinetti and M. Morelli, "An Initial Ranging Scheme for the IEEE 802.16 OFDMA Uplink," in *IEEE Transactions on Wireless Communications*, vol. 11, no. 9, 2012, pp. 3204–3215.
- [3] H. Minn, V. Bhargava, and K. Letaief, "A robust timing and frequency synchronization for OFDM systems," *IEEE Transactions on Wireless Communications*, vol. 2, no. 4, pp. 822–839, July 2003.
- [4] S. D. Choi, J. M. Choi, and J. H. Lee, "An initial timing offset estimation method for OFDM systems in rayleigh fading channel," in *IEEE 64th Vehicular Technology Conference*, September 2006, pp. 1–5.
- [5] E. Zhou, X. Hou, Z. Zhang, and H. Kayama, "A preamble structure and synchronization method based on central-symmetric sequence for OFDM," in *IEEE Vehicular Technology Conference*, 2008, pp. 1478–1482.
- [6] Y. Tsai and G. Zhang, "Time and frequency synchronization for 3GPP long term evolution systems," in *IEEE 65th Vehicular Technology Conference*, April 2007, pp. 1727–1731.
- [7] S. P. Lloyd, "Least squares quantization in PCM," in *IEEE Transactions on Information Theory*, vol. 2, no. 28, 1982, pp. 129–137.
- [8] M. Morelli and U. Mengali, "An improved frequency offset estimator for OFDM applications," in *Communication Theory Mini-Conference*, June 1999, pp. 106–109.