Multiplication-Free Detection Algorithm of the Primary Synchronization Signal in LTE

Mohammad H. Nassralla*, Hassan Ayoub, Naeem Akl§, Rida Jichi‡ and Mohammad M. Mansour*, *Senior Member, IEEE*
American University of Beirut, Lebanon*, Arm, UK‡, Qualcomm NJ, USA§

***Abstract*—Frame synchronization is an important functionality that should be supported in the design of an LTE baseband receiver. Detecting the start of a frame is regularly repeated during transmission and to identify the cells of the network. Typically, the synchronization is performed using correlation-based methods that incur a large number of multiplications and thus increase the power consumption and hardware complexity. Approaches to reduce the rate of multiplications come at the expense of reduced performance of the algorithm. In this paper, we present a *multiplication-free* synchronization algorithm that is based on K-means clustering and distributed arithmetic and uses the primary synchronization signal (PSS). We also show through simulations that the immense improvement in power consumption and hardware complexity does not entail a degradation in performance compared to current synchronization techniques.**

***Index Terms*—3GPP LTE, PSS, synchronization, K-means clustering, distributed arithmetic**

I. Introduction

In the Long-Term Evolution (LTE) standard, users access the wireless communication link through orthogonal frequency division multiple access (OFDMA). The user is required to stay in synchronization with the transmitting base station (eNodeB) in the serving cell. This implies that the user should identify the cell identity, detect the start of the frame, match the frequency of the allocated subcarrier and detect the start of the transmitting signals. Of interest to us is frame synchronization. This is a non-trivial problem as the user might be in continuous motion, crossing different cell borders, subject to varying frequency offsets, etc. which necessitates further action by the user equipment (UE) to sustain the orthogonality among the subcarriers and maintain high quality of experience.

The cell search in LTE is performed hierarchically. The primary synchronization sequence (PSS) and the secondary synchronization sequence (SSS) are transmitted for this purpose on the primary and secondary synchronization channels (P-SCH and S-SCH) respectively. In the first stage the UE identifies the physical layer ID known to be one of three identities 0, 1, or 2 corresponding to the three possible cell ID sectors. This is done on the P-SCH. In the second stage the UE identifies the cell group identity on the S-SCH. There are 168 group IDs ranging from 0 to 167. The physical layer cell ID is then defined as $3 \times \textit{cell group ID} + \textit{cell ID sector}$. The three potential physical layer identities within every cell group identity are assigned in the standard to three neighboring cells. This yields the advantage that an edge user at most equidistant to three serving eNodeBs is now capable of identifying the transmitting source even though almost equal power is received on every P-SCH signal. Yet, this burdens the UE as three time correlations for the PSS signal detection are now to be performed by three matched filters on the receiver side. In addition, 168 frequency domain correlations are run to decode the cell group ID for a complete identification of the physical layer cell ID.

The above synchronization is continuous, should run in real-time and includes a large number of complex multiplications (CMs). The process is thus so complex that it has directed research towards reducing the complexity of the correlation process while maintaining good performance in terms of fast and correct synchronization under different conditions.

Different approaches have been followed. In [1] the correlation is done using the FFT at the expense of increased hardware complexity. In [2] the symmetry of the PSS signal allows to do the correlation with only 35 CMs. Then this is done twice for one cell group ID as two of the PSS signals are complex conjugates [2]. [3] also builds on the symmetry of the PSS signal but correlates its two halves together. In [4] the correlation is performed with only 24 CMs by making use of the repetition of the elements within the PSS sequence. The periodicity of the cyclic prefix in [5] allows the detection of the start of the OFDM symbols using lagged auto-correlation.

In this paper we advise a novel synchronization algorithm using the PSS signal. The algorithm builds on [6] which introduced K-means clustering of the PSS signal as a method to reduce the complexity of the synchronization to 6 CMs. This paper illustrates how K-means clustering can be combined with distributed arithmetic in order to reduce the complexity to zero CM. Thus the algorithm maintains low power consumption and reduced hardware complexity. On the other hand, it also achieves high synchronization performance in terms of the signal acquisition time and the immunity to large carrier frequency offsets. The paper is organized as follows: Section II describes the frame structure in LTE and the PSS design. Section III summarizes the application of K-means clustering to the synchronization problem as part of the algorithm design. Section IV adds the distributed arithmetic component to the algorithm. The simulation results are presented in Section V. Section VI concludes the paper.

II. Frame Structure and PSS Design in LTE

In LTE, six bandwidths ranging from 1.4 MHz to 20 MHz are supported. The spectrum is divided into 15 kHz wide subcarriers arranged into resource blocks (RBs). One RB spans 12 subcarriers and extends over one time slot of length 0.5ms. Scheduling is done on basis of two RBs within one subframe (1ms). Ten subframes constitute the radio frame in LTE, and the minimum number of RBs is six.

Every time slot corresponds to seven OFDM symbols in the normal mode and six OFDM symbols in the extended mode. The two modes refer to the length of the cyclic prefix (CP), which is the guard time interval inserted at the start of every OFDM symbol to avoid intersymbol interference. Independent of the mode, the PSS signal is transmitted in the last OFDM symbol of the first and eleventh slot of every radio frame. Along frequency, the PSS signal occupies the middle 62 subcarriers of the six central RBs regardless of the bandwidth.

The PSS signals are based on Zadoff-Chu (ZC) sequences characterized by their constant amplitude and orthogonality among their cyclically shifted versions. Such sequences are thus useful for synchronization purposes. The ZC sequence is given by:

$$d_u(n) = e^{-j\frac{\pi u n(n+1)}{L}},\ 0 \le n \le L-1, \tag{1}$$

The subindex u is the ZC root index determined by the cell ID sector and its values are selected to be relatively prime to L. L is the ZC length chosen to be 63. For instance, in LTE u takes the values of 25, 29 and 34. The PSS signal is obtained by applying the Inverse Fast Fourier Transform (IFFT) of length $N = 64$ to the ZC sequence and is given by:

$$s_u(n)=\frac{1}{N}\sum_{k=0}^{N-1} d_u\left(k-1\right) e^{j2\pi nk/N},\ n=0,1,\cdots,N-1. \tag{2}$$

A CP is appended to the PSS signal before transmission.

An important property is the inter-sequence conjugate symmetry of the PSS signal. The fact that two PSS sequences with two subindices u and $L-u$ are conjugate symmetric is true also for the PSS signals in (2). Note the selected values of u (29 = L-34).

$$s_u\left(n\right) = s^*_{L-u}\left(n\right) \tag{3}$$

The above symmetry property is important because it can be exploited at the receiver side to reduce the number of CMs encountered in matched-filtering the received signal. To see this, note that the PSS signal is both modulated by the channel upon transmission and corrupted by Gaussian noise. The baseband received signal $r(n)$ can be expressed as:

$$r(n) = c\sum_{m=0}^{M-1} h(m)s_u(n-\theta-m) + z(n), \tag{4}$$

where h is the channel response, M is the number of resolvable paths, z is Gaussian noise, θ is a timing offset and c is function of the carrier frequency offset. As there are three possible PSS signals corresponding to the three possible values of u, the received signal is matched-filtered three times. The sample output for the matched filter of subindex u becomes:

$$\left|y_u\left(m\right)\right|^2 = \left|\sum_{n=0}^{N-1} r\left(n+m\right)s^*_u\left(n\right)\right|^2,\ 0 \le m \le N-1 \tag{5}$$

Once $|y_u\left(m\right)|^2$ exceeds a preset threshold λ, the PSS signal of subindex u is thus detected. The index m at which $|y_u\left(m\right)|^2$ is maximum indicates the start of the PSS signal. Running (5) for the 3 filters requires $3(N-1)$ complex additions (CAs) and $3N$ CMs for every received sample. However, from (3) the individual multiplications within every CM for two of the three sequences are identical. The total number of CMs thus drops to $2N$.

III. Clustering

To further reduce the number of CMs, we alter the sequences to be correlated with the received samples $r(n)$ without significantly degrading the performance of the detection algorithm of the PSS signal. By inspecting (5) we note that the sequence $r(n)$ is correlated against saved samples of $s_u(n)$ for the different indices u. Correlating $r(n)$ with an exact version of sequence $s_u(n)$ allows to detect of the PSS signal at the expense of excessive computations in terms of CMs. Instead, we could think of a more efficient representation of sequence $s_u(n)$ that preserves the detection capability but cuts down the complexity.

In particular, we cluster sequence $S = \{s_u(n), 0 \le n \le N-1\}$ into K clusters, where $K < N$. We partition $s_u(n)$ into K sets $S_k, 1 \le k \le K$ where $S = \cup_{k=1}^{K} S_k$. For each set S_k a centroid μ_k is computed. We then replace $s_u(n)$ in (5) with μ_l where $l = \arg_k S_k | s_u(n) \in S_k$. The correlation metric in (5) is approximated as:

$$\left|y_u\left(m\right)\right|^2 \approx \left|\sum_{k=0}^{K-1}\mu^*_k \sum_{\substack{n=0\\ s_u(n)\in S_k}}^{N-1} r\left(n+m\right)\right|^2,\ 0 \le m \le N-1 \tag{6}$$

Given the above form, by using a look-up table to store the μ_k's for every subindex u, a multiplexer and K accumulators, the number of CMs per computation of $y_u(m)$ for specific values of m and u drops from N to K. This is a clear indication of a sharp reduction in the complexity of the detection algorithm of the PSS signal, particularly for small values of K.

To do the partitioning of S we employ K-means clustering with uniform sample weighting [7]. Similar samples $s_u(n)$ of S are grouped together, and similarity is measured in terms of minimizing the Euclidean distance of the samples to the corresponding centroid [8]. Note that clustering is done once and offline. A design parameter that needs to be determined is K. We choose $K = 6, 8$ for which performance of the detection algorithm is highly maintained even under low signal-to-noise (SNR) ratio of -10dB and a large frequency offset of ±5 ppm as verified through MATLAB simulations.

IV. Distributed Arithmetic

The objective of this section is to show how we can eliminate all the multiplications incurred when detecting the PSS signal through correlation. We employ the distributed arithmetic algorithm described in [9]. Distributed Arithmetic (DA) is a method that allows performing a dot product of a variable vector with a constant vector of values by exploiting storage elements in the form of Look-up Tables (LUTs), implemented as ROM, instead of multiplication execution units. The idea behind DA is to store in LUT entries all the possible sums of elements of the constant vector. These entries are later addressed by bits from the arriving variable vector values, and read into a ROM Accumulate (RAC) structure. It should be noted that the application of DA here is only efficient because we limit the number of clusters to at most $K = 8$. Otherwise DA will have high memory requirements and it becomes unattractive.

This section is organized as follows. DA is targeted for dot products of real vectors. However, synchronization via the PSS signal involves CMs as in (6). We first present a preliminary description of an efficient CM scheme. We then illustrate DA on a dot product of two real vectors. Combining the two schemes (efficient CM and DA), we illustrate how full DA can be employed for PSS synchronization. This will in turn motivate the architecture that we use for the DA part of our multiplication-free clustering-and-DA detection scheme.

A. *Efficient Complex Multiplication*

Consider two complex numbers A and B. Let A_R and B_R be the real parts of A and B respectively, while A_I and B_I be their imaginary parts. The straightforward CM scheme involves four real multiplications: $AB = (A_R B_R - A_I B_I) + j(A_R B_I + A_I B_R)$. A more efficient CM is performed using three real multiplications:

$$\begin{aligned} AB = {} & [B_R(A_R - A_I) + A_I(B_R - B_I)] \\ & + j[A_R(B_R + B_I) - B_R(A_R - A_I)] \end{aligned} \quad (7)$$

B. *Distributed Arithmetic for Dot Products of Real Vectors*

Let $\overrightarrow{c} = [A, B, C, D, E, F]$ be a vector of $K = 6$ **constant real** numbers. We illustrate the DA method for $K = 6$, while its application for the case $K = 8$ is straightforward. Let $\overrightarrow{v} = [X, Y, Z, W, U, V]$ be a vector of **variable real** numbers. Each of $[X, Y, Z, W, U, V]$ is 16-bit valued, where $[X_0, X_1, \ldots, X_{15}]$ for example refer to the 16 bit values of X with X_0 being the least significant bit and X_{15} being the sign bit. Thus we use Q15 fixed-point representation of numbers where for instance X is given by:

$$X = -X_{15}.2^{15} + \sum_{i=0}^{14} X_i.2^i \quad (8)$$

We are interested in computing the dot product $\overrightarrow{c}.\overrightarrow{v} = AX + BY + CZ + DW + EU + FV$. This can be expressed as:

$$\begin{aligned} \overrightarrow{c}.\overrightarrow{v} = {} & -[AX_{15} + BY_{15} + CZ_{15} + DW_{15} \\ & + EU_{15} + FV_{15}].2^{15} \\ & + \sum_{i=0}^{14}[AX_i + BY_i + CZ_i + DW_i + EU_i + FV_i].2^i \end{aligned} \quad (9)$$

Since the bit values are either one or zero, each of $[A, B, C, D, E, F]$ is included or excluded accordingly in the sums within the brackets. At any bit location i, the bit-vector $[X_i, Y_i, Z_i, W_i, U_i, V_i]$ can assume one of 64 values, and therefore the terms within every bracket will add up to one of 64 possible summation values that can be computed beforehand and stored in a look-up table (LUT). Actually, the bit-vector $[X_i, Y_i, Z_i, W_i, U_i, V_i]$ is an address to where the sum $AX_i + BY_i + CZ_i + DW_i + EU_i + FV_i$ is stored within the LUT. The DA algorithm is illustrated in Figure 1. Multiplications by powers of 2 are simply bit-shifts.

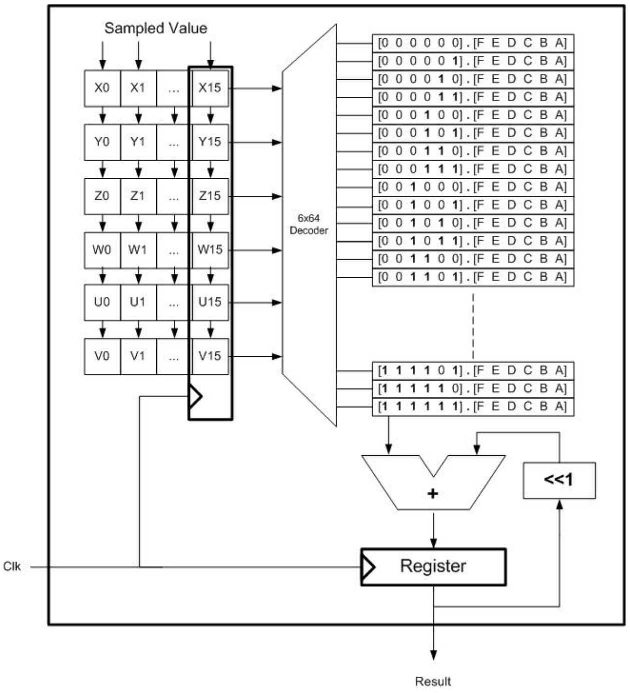
Sampled Value
X0
X1
...
X15
Y0
Y1
...
Y15
Z0
Z1
...
Z15
W0
W1
...
W15
U0
U1
...
U15
V0
V1
...
V15
6x64
Decoder
[0 0 0 0 0 0].[F E D C B A]
[0 0 0 0 0 1].[F E D C B A]
[0 0 0 0 1 0].[F E D C B A]
[0 0 0 0 1 1].[F E D C B A]
[0 0 0 1 0 0].[F E D C B A]
[0 0 0 1 0 1].[F E D C B A]
[0 0 0 1 1 0].[F E D C B A]
[0 0 0 1 1 1].[F E D C B A]
[0 0 1 0 0 0].[F E D C B A]
[0 0 1 0 0 1].[F E D C B A]
[0 0 1 0 1 0].[F E D C B A]
[0 0 1 0 1 1].[F E D C B A]
[0 0 1 1 0 0].[F E D C B A]
[0 0 1 1 0 1].[F E D C B A]
[1 1 1 1 0 1].[F E D C B A]
[1 1 1 1 1 0].[F E D C B A]
[1 1 1 1 1 1].[F E D C B A]
+
<<1
Clk
Register
Result

Fig. 1. Correlation with the PSS sequence applied using the DA algorithm.

As a side note, for each bit location of the values in the variable vector, one LUT access is required. If DA is implemented serially, a total of n serial accesses is required for the product result to converge, where n is the bit-size of values in the variable vector. If storage elements are abundant and computation time is critical, DA can be implemented fully parallel whereby the n LUT accesses can all be done simultaneously, which requires replicating the LUTs for n times, achieving result convergence time of one clock cycle.

C. *Selected Architecture*

Let $\overrightarrow{A} = [A^{(1)}, A^{(2)}, A^{(3)}, A^{(4)}, A^{(5)}, A^{(6)}]$ be the complex vector holding the conjugates of the $K = 6$ centroids, i.e. $\mu_k^*, 0 \leq k \leq K - 1$. Denote by $\overrightarrow{B} = [B^{(1)}, B^{(2)}, B^{(3)}, B^{(4)}, B^{(5)}, B^{(6)}]$ the vector of the six complex summations $\sum_{n=0, s_u(n) \in S_k}^{N-1} r(n+m)$ in (6) corresponding to the six values of k, $0 \leq k \leq K - 1$. Let $\overrightarrow{A}_R$ and $\overrightarrow{A}_I$ be two real vectors holding respectively the real and imaginary parts of $\overrightarrow{A}$ element-wise. $\overrightarrow{B}_R$ and $\overrightarrow{B}_I$ are similarly derived from $\overrightarrow{B}$.

From (7), the complex dot product in (6) can be computed from the following dot products of real vectors:

- $\overrightarrow{B}_R(\overrightarrow{A}_R - \overrightarrow{A}_I)$
- $\overrightarrow{A}_I(\overrightarrow{B}_R - \overrightarrow{B}_I)$
- $\overrightarrow{A}_R(\overrightarrow{B}_R + \overrightarrow{B}_I)$

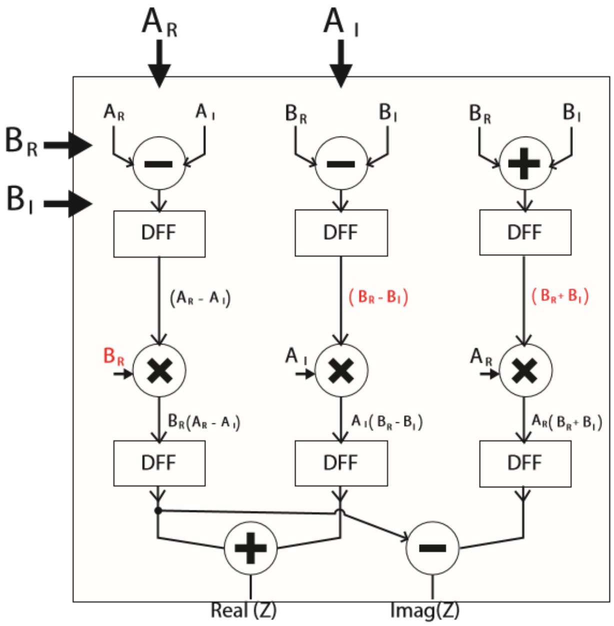

Fig. 2. Complex multiplication involving only three real multiplications.

In our implementation, we choose to compute these three dot products in parallel. These dot products along the other connections are shown in figure 2. Each of the dot products follows the architecture of figure 1. Within each product the LUT accesses are serial, so a total of three LUTs are needed. Each LUT has a size of $2^K = 64$ memory elements.

V. Performance Analysis and Simulation Results

We use the same simulation parameters as described in [2]. Thus we assume that the channel is 6-path typical urban (TU) with low SNR of $-10\,\mathrm{dB}$. The speed of the UE is $3\,\mathrm{km/hr}$, and we consider two search scenarios: initial cell search and neighbor cell search. The two scenarios are modeled by $\pm 5\,\mathrm{ppm}$ and $\pm 0.1\,\mathrm{ppm}$ frequency offsets respectively. For the static scenario we assume a single-path AWGN channel with zero frequency offset. First MATLAB simulation results on the performance of the detection algorithm are presented. We then show that the results hold true for the Q15 fixed-point implementation of DA.

A. Performance of the Proposed Detection Algorithm

To test the performance of the described synchronization algorithm, we again use the same performance metric as in [2]. Particularly we examine the miss-detection probability and the cumulative distribution of the obtained acquisition time, where lower miss rate at a given SNR and less acquisition time indicate better performance. For comparison purposes we show the results for the following synchronization algorithms:

- A brute-force matched-filtering (MF) search algorithm (OPTIMAL) for a PSS sequence (2) of $N = 64$ samples. This corresponds to 64 CMs per incoming PSS sample. Equivalently, Popovic's algorithm [2] (POPOVIC) performs an identical MF operation as OPTIMAL for $N = 64$-long PSS signal for only 33 CMs.
- Our proposed algorithm (CLUSTER8-USx2/ CLUSTER6-USx2) at 8 and 6 clusters for $N = 64$, upsampled at the receiver by a factor of two and 0 CMs. Note that upsampling by x2 introduces no CMs because each interpolated sample is obtained as an average of two original samples, which is simply their sum shifted one bit to the right.
- Q15-Fixed point implementation of DA at 6 clusters for $N = 64$ and 0 CMs.

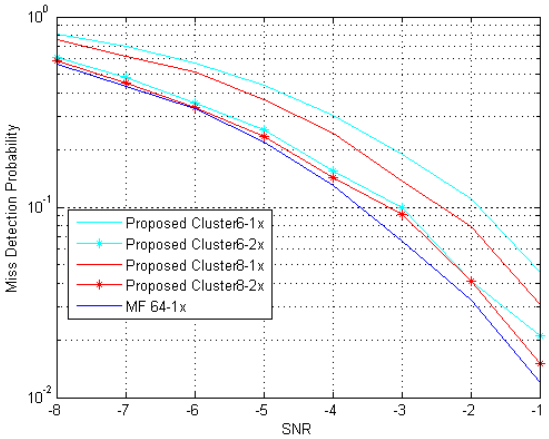

Fig. 3. Probability of miss-detection versus SNR in dB for a false alarm probability of 0.1.

Figure 3 shows the variation of the miss-probability versus the SNR for MF (Optimal/Popovic) and the proposed algorithm for $K = 6, 8$ and two upsampling factors 1x and 2x in a static scenario. The false alarm probability is fixed at 0.1. Clearly as the SNR increases there is a higher chance to detect the PSS so the curves are decreasing. Curves shifted to the right indicate a higher SNR requirement for the same detection performance.

The distribution of the acquisition time for OPTIMAL (or POPOVIC), CLUSTER8 and CLUSTER6 is shown in Figure 4. All the curves are positively sloped since acquisition is more likely within a wider time window. Moreover, for all algorithms the acquisition time is statistically larger for the higher frequency offset ($\pm 5\,\mathrm{ppm}$). Again the performance of CLUSTER8 and CLUSTER6 is close to OPTIMAL. The gap from OPTIMAL becomes negligible for an upsampled (2xN=128)-PSS sequence. It should be also noted that if MF is applied on an upsampled PSS signal, its performance remains almost unaltered as shown in Figures 3 and 4.

B. Q15 Fixed-point implementation

The correlation algorithm is implemented using clustering and DA under Q15 fixed-point representations to detect the PSS. Simulation results show minimal effect on the miss detection probability as illustrated in Figure 6. First, we note that similar curves are obtained for the clustering algorithm as those in [6], which indicates that the DA method cuts down the complexity at minimal performance degradation. As compared to MF, clustering + DA totally eliminates the CMs per PSS sample at a minimal increase in the SNR requirement for the

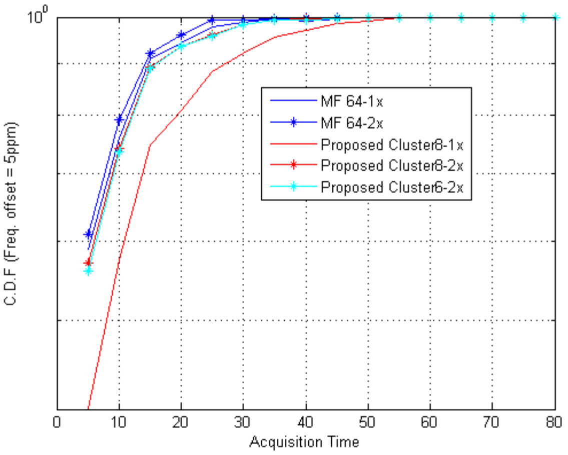

Fig. 4. CDF of acquisition time for SNR = -5dB at 5ppm and false alarm probability of 0.1.

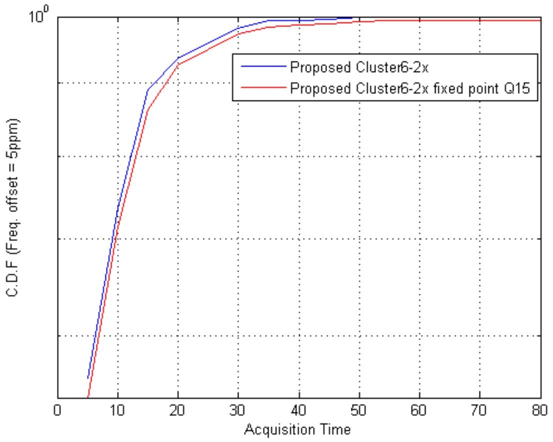

Fig. 5. Effect of the fixed point implementation of DA on the CDF of acquisition time for SNR = -5dB at 5ppm and false alarm probability of 0.1.

same error performance.

Similar results hold for the CDF of the acquisition time when the correlation algorithm is implemented under clustering and DA at frequency offsets ± 5 ppm as shown in Figure 5. This is also achieved at zero CMs. This excellent performance-complexity tradeoff is obtained by exploiting the advantage of the reduced complexity of clustering and the intelligent architecture of distributed arithmetic.

VI. Conclusion

A novel frame synchronization algorithm in LTE using the PSS signal is presented. The algorithm is based on K-means clustering and distributed arithmetic. It involves no multiplications as opposed to correlation-based techniques, which immensely cuts the power consumption and reduces the hardware complexity. In addition it achieves high synchronization performance compared to state-of-the-art solutions. The impact of varying the number of clusters and upsampling the received signal are examined in simulation for different search scenarios and frequency offsets.

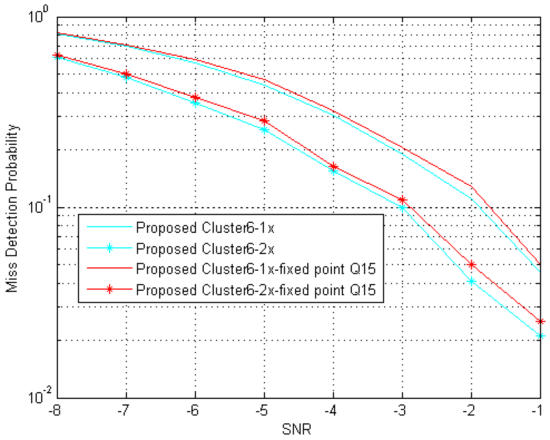

Fig. 6. Effect of the fixed point implementation of DA on the probability of miss-detection versus SNR in dB for a false alarm probability of 0.1.

References

[1] C. Cheng and K. K. Parhi, "High-throughput vlsi architecture for fft computation," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 54, pp. 863–867, Oct 2007.

[2] B. M. Popovic and F. Berggren, "Primary synchronization signal in e-utra," in *2008 IEEE 10th International Symposium on Spread Spectrum Techniques and Applications*, pp. 426–430, Aug 2008.

[3] A. Dammann, C. Mensing, and S. Sand, "On the benefit of location and channel state information for synchronization in 3gpp-lte," in *2010 European Wireless Conference (EW)*, pp. 711–717, April 2010.

[4] Y. Yang, W. Che, N. Yan, X. Tan, and H. Min, "Efficient implementation of primary synchronisation signal detection in 3gpp lte downlink," *Electronics Letters*, vol. 46, pp. 376–377, March 2010.

[5] W. Xu and K. Manolakis, "Robust synchronization for 3gpp lte system," in *2010 IEEE Global Telecommunications Conference GLOBECOM 2010*, pp. 1–5, Dec 2010.

[6] M. H. Nassralla, M. M. Mansour, and L. M. A. Jalloul, "A low-complexity detection algorithm for the primary synchronization signal in lte," *IEEE Transactions on Vehicular Technology*, vol. 65, pp. 8751–8757, Oct 2016.

[7] A. Ben Ayed, M. Ben Halima, and A. M. Alimi, "Survey on clustering methods: Towards fuzzy clustering for big data," in *2014 6th International Conference of Soft Computing and Pattern Recognition (SoCPaR)*, pp. 331–336, Aug 2014.

[8] G. Ahalya and H. M. Pandey, "Data clustering approaches survey and analysis," in *2015 International Conference on Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE)*, pp. 532–537, Feb 2015.

[9] S. A. White, "Applications of distributed arithmetic to digital signal processing: a tutorial review," *IEEE ASSP Magazine*, vol. 6, pp. 4–19, July 1989.