Implementation of LTE System using FIFO Techniques

V. N. Priyadarsani¹, G. S. Babu²
M.Tech Student¹, Assistant Professor²
Department of Electronics & Communication Engineering
D.N.R College of Engineering and Technology, Bhimavaram, India

Abstract:
A novel architecture for efficient time and frequency synchronization, applied to the long-term evolution (LTE) standard, is proposed. For symbol timing, we propose applying a symbol-folding method on top of the sign-bit reduction technique, leading to a novel algorithm for the cyclic prefix-type recognition in LTE. Following the symbol timing, the fractional carrier frequency offset is estimated and compensated using an adaptive gain loop, which allows for high-accuracy compensation in a short interval. In the frequency domain, for cell search, we propose a sign-bit reduction technique on top of the matched filter method for the primary synchronization signal detection. In addition, we propose the sign-bit maximum-likelihood sequence detection algorithm for the secondary synchronization signal analysis. These methods result in >90% hardware reduction in the cell search compared with the state-of-the-art design. Moreover, possible structures for the frequency tracking in LTE are investigated leading to an experimental comparison, used to choose the best hardware-efficient structure.

I. INTRODUCTION

The long-term evolution (LTE), introduced by the third-generation partnership project (3GPP), is an emerging standard for high-speed wireless communications. This standard benefits from the orthogonal frequency division multiplexing (OFDM) technology in the downlink, which provides several advantages, such as high and width efficiency, robustness to the multipath fading, and simplicity of

A typical method to implement the coarse synchronization step in the pre-fast Fourier transform (FFT) domain is based on an autocorrelation that estimates the OFDM symbol beginning and the fractional CFO using the cyclic prefix (CP) of the OFDM symbols. This method incurs a high computational complexity especially for long symbols. A simplified approach is, where the sign bit of the samples are used, replacing the 8-bit multipliers by XNOR gates. Although the proposed method is effective for the largest inverse FFT (IFFT) size (N IFFT) in LTE, its performance deteriorates considerably for smaller IFFT sizes, which are very common in many modes and practical standards. This is due to the fact that for smaller IFFT sizes, the shorter CP length makes the detection more vulnerable to natural perturbations. Moreover, this method required a prior knowledge of the CP length while the CP-type is not known at this stage and should be recognized at the same time.

<table>
<thead>
<tr>
<th>N IFFT</th>
<th>Bandwidth (MHz)</th>
<th>Occupied Subcarriers</th>
<th>Normal CP</th>
<th>Extended CP</th>
</tr>
</thead>
<tbody>
<tr>
<td>128</td>
<td>1.4</td>
<td>76</td>
<td>10/9</td>
<td>32</td>
</tr>
<tr>
<td>256</td>
<td>3</td>
<td>151</td>
<td>20/18</td>
<td>64</td>
</tr>
<tr>
<td>512</td>
<td>5</td>
<td>301</td>
<td>40/56</td>
<td>128</td>
</tr>
<tr>
<td>1024</td>
<td>10</td>
<td>601</td>
<td>80/72</td>
<td>256</td>
</tr>
<tr>
<td>2048</td>
<td>20</td>
<td>1201</td>
<td>160/144</td>
<td>512</td>
</tr>
</tbody>
</table>

A method for CP-type detection where two FFT blocks are employed to check for both possible CP types resulting in a high computational complexity, while failing in nearly half of the cases. Our proposed architecture detects the CP-type through a practical method. Moreover, we apply symbol folding on top of the sign-bit reduction technique leading to a low-complexity and high-performance symbol boundary and fractional CFO estimation architecture for all LTE defined transmission modes. This architecture avoids any multiplier or CORDIC block.
providing 80% area reduction in the coarse frequency synchronization comparing with the state-of-the-art (Table IX). The second step of the synchronization process is cell search, i.e., as soon as a connection with the network is established, it is required to detect the cell ID group and the sector ID. Other information, such as integer CFO and LTE frame timing are also determined as a by-product. Cell search is achieved by the detection and analysis of predefined primary synchronization signal (PSS) and secondary synchronization signal (SSS) sequences. In general, comparison between the received data and the expected pattern can determine the PSS position. Although these methods provide reasonable performance, they normally result in a very high computational complexity. The hardware complexity by simplifying the analog-to-digital converter (ADC) in the system is decreased. In this paper, we focus on the rest of the architecture, i.e., the architecture without considering the ADC as a part of it. We reduce the hardware complexity of the proposed architecture by 96% compared with the state-of-the-art design while maintaining the required performance. Once coarse synchronization and cell search are performed, the fine synchronization step, through frequency tracking, needs to be done. The fine synchronization estimates and compensates the residual CFO of the received symbols in the post-FFT domain. In this paper, all possible combinations of existing methods for the fine synchronization in OFDM-based systems are investigated. To provide a comprehensive comparison, all possible structures with different combinations of calculation methods are evaluated in various situations. Moreover, a set of comparison parameters are introduced to compare the structures both experimentally and analytically.

### TABLE II

**Computational Complexity of PSS Detection Methods**

<table>
<thead>
<tr>
<th>no.</th>
<th>Method</th>
<th># of Additions</th>
<th># of Multipliers</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>CCU/AC</td>
<td>57289</td>
<td>24352</td>
</tr>
<tr>
<td>1</td>
<td>MDF/DF</td>
<td>16967</td>
<td>6138</td>
</tr>
<tr>
<td>2</td>
<td>MDF/DF (proposed)</td>
<td>8155</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>CCU/AC</td>
<td>57 (28 x 2 + 1 + 1)</td>
<td>2 (12 x 2 + 2)</td>
</tr>
<tr>
<td>4</td>
<td>MDF/DF</td>
<td>17 (8 x 3 + 1)</td>
<td>6 (3 x 8)</td>
</tr>
<tr>
<td>5</td>
<td>MDF/DF (proposed)</td>
<td>9 (4 x 2 + 1)</td>
<td>0</td>
</tr>
</tbody>
</table>

The analytical contributions of this paper include increasing the performance of the coarse synchronization block, decreasing the hardware complexity in the cell search, and introducing a comprehensive yet experimental comparison among various possible solutions for frequency tracking.

### FRAME STRUCTURE IN LTE STANDARD

LTE standard supports a scalable transmission bandwidth ranging from 1.4 to 20 MHz, which translates to FFT sizes of 128–2048 (Table I). Regardless of the transmission mode, the synchronization channels occupy a 1.25-MHz central frequency band. This frequency band corresponds to 72 central subcarriers, including 62 subcarriers carrying the synchronization signals and 5 null subcarriers reserved at both ends. Synchronization signals, including the PSS and SSS signals, are located in two consecutive symbols at the beginning and middle of a frame. According to the 3GPP-LTE standard [1], each downlink frame (10 ms) is divided into 10 subframes, each 1-ms long. Each subframe consists of two 0.5-ms slots, each containing six or seven OFDM symbols. The PSS signal is always mapped to the last symbol in slots 0 and 10, while the SSS signal is mapped to the symbol ahead of the PSS signal in the same slot. Detection and analysis of PSS and SSS signals are required to detect the cell ID. To connect to the network successfully, a user equipment requires to detect the cell ID that is composed of cell ID group (N(1) ID ) and sector ID (N(2) ID ) as follows:

\[
N_{\text{ID}}^{\text{cell}} = 3N_{\text{ID}}^{(1)} + N_{\text{ID}}^{(2)}, \quad 0 \leq N_{\text{ID}}^{(1)} \leq 167, \quad 0 \leq N_{\text{ID}}^{(2)} \leq 2.
\]

#### A. PSS Sequences

PSS sequences are constructed on the basis of the Zad Off–Chu (ZC) sequence, inheriting special properties for detection. Three patterns of the PSS sequence are defined in the frequency domain, which are ZC sequences with the length of 63, without any dc subcarriers as follows:

\[
d_{u}(n) = \begin{cases} 
\left(\frac{\pi u(n + 1)}{63}\right), & n = 0, 1, \ldots, 30 \\
\left(\frac{\pi u(n + 1)(n + 2)}{63}\right), & n = 31, 32, \ldots, 61
\end{cases}
\]

where \(n\) is the number of subcarriers and \(d_{u}(n)\) is the \(n\)th subcarrier of the PSS sequence with \(u\) as the root index. Each root index specifies one \(N(2) \text{ ID}\), i.e., \(N(2) \text{ ID} = 0\) when \(u = 25\), \(N(2) \text{ ID} = 1\) when \(u = 29\), and \(N(2) \text{ ID} = 2\) when \(u = 34\).

#### B. SSS Sequences

SSS sequences are combinations of three 31-length binary sequences, i.e., \(s(n)\), \(c(n)\), and \(z(n)\), where \(c(n)\) and \(z(n)\) are scrambling sequences and \(s(n)\) is a basic sequence useful for distinguishing the cell ID group. The SSS subcarriers, defined in the 3GPP-LTE standard, are as follows [1]:

\[
\begin{align*}
\text{subframe 0:} & \quad d_{0}(2n) = s_{m_{0}}(n) \cdot c_{0}(n) \\
& \quad d_{0}(2n + 1) = s_{m_{1}}(n) \cdot c_{1}(n) \cdot s_{m_{0}}(n) \\
\text{subframe 5:} & \quad d_{1}(2n) = s_{m_{0}}(n) \cdot c_{0}(n) \\
& \quad d_{1}(2n + 1) = s_{m_{1}}(n) \cdot c_{1}(n) \cdot s_{m_{1}}(n)
\end{align*}
\]

where \(d_{0}(n)\) and \(d_{1}(n)\) are the binary values of the \(n\)th subcarriers of the SSS sequence in the zeroth and fifth subframes, respectively. Each pair of \((m_{0}, m_{1})\) corresponds to one cell ID group, and \(s_{m_{0}}(n)\) and \(s_{m_{1}}(n)\) are the cyclically shifted versions of \(s(n)\) \(s_{m_{i}}(n) = s(n + m_{i} \mod 31)\), \(i = 0, 1\).

In the same way, \(z_{m_{0}}(n)\) and \(z_{m_{1}}(n)\) are generated from \(z(n)\) as follows:

\[
\begin{align*}
\text{zm}_{k}(n) & = z(n + (m_{k} \mod 8) \mod 31), \quad k = 0, 1
\end{align*}
\]

Finally, \(c_{0}(n)\) and \(c_{1}(n)\) are derived from \(c(n)\) sequence as \(c_{0}(n) = z_{--n + N(2) \text{ ID} \mod 31} \cdot c_{1}(n) = z_{--n + N(2) \text{ ID} + 3 \text{\ mod 31}}\).

#### C. Symbols in the Time Domain

Symbols are received in the time domain, carrying an added CP at their beginning. In the LTE systems, while the CP length varies in different transmission modes, the ratio of CP length to symbol length stays the same (Table I). Moreover, in all defined modes and regardless of the MFFT value, the subcarrier spacing \((f_{sc})\) is 15 kHz, while the symbol time duration \((T_{u})\) is 66.67 \(\mu s\). These parameters are related to each other based
On the subcarrier or orthogonality feature in an OFDM system as follows: \( T_u = 1/f_{sc} \). (7) This configuration makes it impossible for the receiver to recognize the IFFT in the pre-FFT stages of synchronization. Therefore, the FFT size in the receiver side should be set to the maximum possible IFFT size, i.e., IFFT = 2048.

II. PROPOSED VLSI ARCHITECTURE

Synchronizer is used when two designs operate in different frequencies. As a design FFT works on parallel data and inputs will come in terms of serial-data. Therefore we need to convert serial data into parallel data to the input of FFT. Input frequency, design input frequency and output frequency are synchronized by using reset, preset and enables. If variation will be high, it is not possible to do using SIPO and PISO. Therefore we are using FIFO based synchronizer for LTE network.

**Figure 3. Existing synchronizer using Serial-In-Parallel-Out and Parallel-in-Serial-out**

Serial-in to Parallel-out (SIPO) Shift Register

The operation is as follows. Let’s assume that all the flip-flops (FFA to FFD) have just been RESET (CLEAR input) and that all the outputs QA to QD are at logic level “0” i.e., no parallel data output. If a logic “1” is connected to the DATA input pin of FFA then on the first clock pulse the output of FFA and therefore the resulting QA will be set HIGH to logic “1” with all the other outputs still remaining LOW at logic “0”. Assume now that the DATA input pin of FFA has returned LOW again to logic “0” giving us one data pulse or 0-1-0.

**Figure 4. Serial-in Parallel Out**

The second clock pulse will change the output of FFA to logic “0” and the output of FFB and QB HIGH to logic “1” as its input D has the logic “1” level on it from QA. The logic “1” has now moved or been “shifted” one place along the register to the right as it is now at QA.

When the third clock pulse arrives this logic “1” value moves to the output of FFC (QC) and so on until the arrival of the fifth clock pulse which sets all the outputs QA to QD back again to logic level “0” because the input to FFA has remained constant at logic level “0”.

The effect of each clock pulse is to shift the data contents of each stage one place to the right, and this is shown in the following table until the complete data value of 0-0-0-1 is stored in the register. This data value can now be read directly from the outputs of QA to QD.

Then the data has been converted from a serial data input signal to a parallel data output. The truth table and following waveforms show the propagation of the logic “1” through the register from left to right as follows.

<table>
<thead>
<tr>
<th>Clock Pulse No</th>
<th>QA</th>
<th>QB</th>
<th>QC</th>
<th>QD</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Note that after the fourth clock pulse has ended the 4-bits of data (0-0-0-1) are stored in the register and will remain there provided clocking of the register has stopped. In practice the
input data to the register may consist of various combinations of logic “1” and “0”.

**Parallel-in to Serial-out (PISO) Shift Register**

The Parallel-in to Serial-out shift register acts in the opposite way to the serial-in to parallel-out one above. The data is loaded into the register in a parallel format in which all the data bits enter their inputs simultaneously, to the parallel input pins PA to PB of the register. The data is then read out sequentially in the normal shift-right mode from the register at Q representing the data present at PA to PB. This data is outputted one bit at a time on each clock cycle in a serial format. It is important to note that with this type of data register a clock pulse is not required to parallel load the register as it is already present, but four clock pulses are required to unload the data.

**Figure 5. Parallel-in Serial Out**

As this type of shift register converts parallel data, such as an 8-bit data word into serial format, it can be used to multiplex many different input lines into a single serial DATA stream which can be sent directly to a computer or transmitted over a communications line.

**Figure 6. Proposed synchronizer using Pre-FIFO and POST-FIFO**

**First-in, First-Out (FIFO)**

FIFOs are commonly used in electronic circuits for buffering and flow control between hardware and software. In its hardware form, a FIFO primarily consists of a set of read and writes pointers, storage and control logic. Storage may be SRAM, flip-flops, latches or any other suitable form of storage. For FIFOs of non-trivial size, a dual-port SRAM is usually used, where one port is dedicated to writing and the other to reading. A synchronous FIFO is a FIFO where the same clock is used for both reading and writing. An asynchronous FIFO uses different clocks for reading and writing. Asynchronous FIFOs introduce met stability issues. A common implementation of an asynchronous FIFO uses a Gray code (or any unit distance code) for the read and writes pointers to ensure reliable flag generation. One further note concerning flag generation is that one must necessarily use pointer arithmetic to generate flags for asynchronous FIFO implementations. Conversely, one may use either “leaky bucket” approach or pointer arithmetic to generate flags in synchronous FIFO implementations.

**III. IMPLEMENTATION RESULTS**

**Figure 7. Block Diagram for FFT**
IV. CONCLUSION

In this paper, a time and frequency synchronizer is proposed for the LTE systems. The synchronization scheme in this architecture includes coarse time and frequency synchronization in the time domain followed by cell search and CFO tracking in the SIPO domain. In the PIPO synchronization, a symbol folding method on top of sign-bit reduction and a practical method for CP-type recognition are proposed. Moreover, CFO value is estimated and compensated in an adaptive loop, which benefits from both fast and high accuracy compensation. Implementation results show an 80% hardware reduction in this step. In cell search, the proposed architecture applies resource sharing as well as sign-bit reduction on top of the MF method to reduce the complexity of PSS detection >90% compared with the state-of-the-art. Moreover, the sign-bit MLSD method is proposed to increase the accuracy of SSS analysis while lowering the complexity ∼99%. For frequency tracking, the DA-F-L structure is selected after a comprehensive and experimental investigation and comparison among the possible solutions. In this paper, a version of the proposed architecture is fabricated using 130-nm CMOS technology occupying 0.68 mm2 silicon area.

VI. REFERENCES


