Available online at www.elixirpublishers.com (Elixir International Journal)

Network Engineering

Elixir Network Engg. 59 (2013) 15698-15703

ISSN: 2229-712X

Timing and Frequency synchronization in OFDM

Rakhi Thakur and Kavita Khare
MANIT, Bhopal.

ARTICLE INFO

Article history:
Received: 29 April 2013;
Received in revised form:
3 June 2013;
Accepted: 12 June 2013;

Keywords
Multiple-Input Multiple-Output,
Orthogonal Frequency Division Multiplexing,
Frame Synchronization,
Symbol Timing Synchronization,
FPGA.

ABSTRACT
With the advent of OFDM for WLAN communications, as exemplified by IEEE 802.11a, it has become imperative to have efficient and reliable synchronization algorithms for OFDM WLAN receivers. The main challenges with synchronization deal with the delay spread and frequency offset introduced by the wireless channel. In this paper, research is done into OFDM WLAN synchronization algorithms, and a synchronizer implementation is presented. This synchronizer performs packet detection; frequency offset estimation, and time synchronization.

Introduction

OFDM could be tracked to 1950's but it had become very popular at these days, allowing high speeds at wireless communications [16]. While OFDM has become the core of most 4G-communication systems it was essential to build OFDM synchronizer on a suitable hard ware. The aim of our paper is to implement this system to be suitable for all new communication systems. FPGAs are flexible and reconfigurable integrated circuits, whose functionality is programmed by the designer rather than the device manufacturer. Unlike an Application- Specific Integrated Circuit (ASIC), FPGAs can be reprogrammed multiple times, even after deployment. The high speed, parallel architecture provides complete control over the degree of parallelism in the design, and arithmetic word lengths. This flexibility is a key advantage of FPGAs over traditional Digital Signal Processor (DSP) processors. In our implementation, the emulation time has been made as short as possible [17].

Synchronization Issues with OFDM

With any data communications system, a critical component is the ability for the receiver to detect the transmission of a packet. The effects of the Wireless channel can complicate this, so it is important to have packet detection algorithms, which can account for channel effects. The wireless channel also affects the orthogonality of the sub carriers. If there is an offset between the subcarrier frequencies at the transmitter and the subcarrier frequencies at the receiver, the tones will no longer be orthogonal, and this can cause significant degradation in system performance. To maintain this orthogonality, the transmitter and receiver must be precisely synchronized in terms of frequency. This requires accurate frequency offset calculation at the receiver [2].

Implementation of the Synchronizer

Packet Detection:

The first challenge for the receiver is to detect the packet. One possible algorithm to use is packet detection based on power level. That is, the presence of a packet can be inferred when the signal power exceeds a specific threshold. However, packet detection cannot be done in this manner in wireless systems because of the channel noise and multipath fading, which cause the received power to vary. So another technique is to use signal auto-correlation, taking advantage of the repetition in the preamble, and correlating the received sequence samples with a delayed copy of the sequence, with the delay being equivalent to the length of one symbol. A moving average of this correlation can be taken over a range of one symbol. If L is the number of samples and r_d is d^{th} incoming sample then the average correlation R(d) is given by :-

$$R(d) = \sum_{m=0}^{L-1} (r_{d+m}^{*} r_{d+m+L}) \quad \text{----------3.1}$$

Where r^*_d represents the conjugate of r_d and m is an integer. For such Packet Detector received power P (d) is:

$$P(d) = \sum_{m=0}^{L-1} | r_{d+m+L} |^2 \quad \text{----------3.2}$$

For a threshold (th) value, and ensuring correct functionality during simulations as shown in fig.1 the ratio of square of average correlation and square of received power M (d) should be fall between 0 and 1.

$$M(d) = \frac{|R(d)|^2}{(P(d))^2} \quad \text{----------3.3}$$

Fig 1 shows the basic structure of packet detector in which * block is used for calculating the correlation. Here a delay of 16 is provided for the incoming samples because in the case of the STS symbols, L = 16 samples. The value threshold should be chosen to minimize the incidence of false positive detection, and also the incidence of undetected packets, which occur when the receiver is unable to detect the training sequence. The value of

Tele:
E-mail addresses: rakhi082003@yahoo.co.in

M (d) is then compared with this threshold, and a packet is said to be detected if M (d) > th [3].

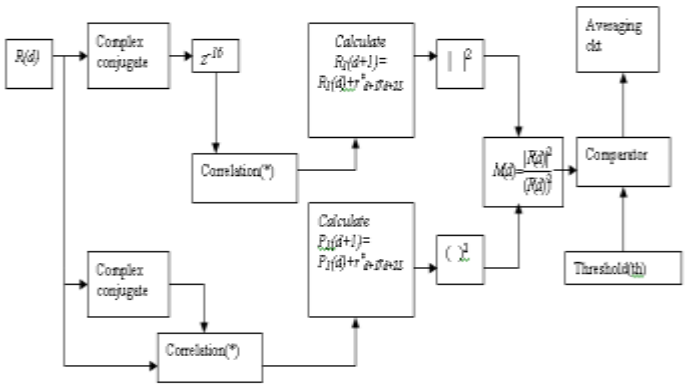

Figure 1: Packet Detector Block Diagram

In accordance with the analysis [3] the threshold should be set at 0.5 × P (d). Because the division operation required in Equation 3.3 can be avoided by choosing a metric threshold level, which is a power of 2. For instance, choosing a threshold value of 0.5 will allow the calculation using a bit shift operation rather than a division operation. The advantage is hardware savings, and the value of 0.5 is actually a very good threshold choice. The output of the packet detection circuit includes a control signal indicating when a packet has been detected, as well as the auto-correlation and power values, which are also used elsewhere in the synchronizer. The packet detection output control signal should only become non-zero after the AGC has completed [16].

Frequency Offset Calculation and Carrier Frequency Synchronization

In OFDM link the sub carriers are perfectly orthogonal if transmitter and receiver use exactly the same frequencies. Any deviation in frequencies causes frequency offset which can introduce inter channel interference (ICI) and inter symbol interference (ISI). Frequency error sensitivity is a weakness of OFDM systems, since small changes in the sub-carrier frequency caused by distortions in either the channel or the receiver can make the sub-carriers loose their orthogonality. Once this occurs, the interference between adjacent sub-carriers becomes significant and the received signal level is reduced [2]. Methodologies for estimating the carrier frequency offset rely on the fact, if two identical samples are transmitted over the channel, the phase difference between them at the receiver is proportional to the frequency offset, and also proportional to the separation between the two transmission times. Specifically, for a frequency offset of Δf the magnitude of the phase offset at any time t is [5]:

$$\phi = 2\pi t \Delta f \quad \text{----------3.4}$$

This phase difference can be calculated by observing incoming samples separated by one symbol length. For the STS, the phase difference can be extracted from the autocorrelation value R(d) in Equation 3.1. R(d) can be expressed as:

$$R(d) = e^{-j2\pi L\Delta f} \sum_{m=0}^{L-1} | r_{d+m} |^2 \quad \text{----------3.5}$$

The relationship between the ϕ value in Equation 3.4 and the term R (d) in Equation 3.5 is given by:

$$\phi = \angle R(d) \quad \text{----------3.6}$$

In this case, L = 16 samples, for a total time difference of 16Ts, where Ts is the sample period, 50 ns in 802.11a. Thus, if the phase difference value can be determined, an estimate of the frequency offset can be calculated as:

$$\Delta f = \frac{\angle R(d)}{2\pi \times 16T_s} \quad \text{----------3.7}$$

The value R(d) will fall between π and − π in 802.11a, and thus the range of possible frequency offset values is

$$-625\,kHZ \le \Delta f \le 625\,kHz$$

Because of the improved precision, performing the calculation with a 64-sample autocorrelation is referred to as fine frequency offset estimation, while the method using a 16-sample auto-correlation is referred to as coarse frequency offset estimation. Because of the limited range in the 64-sample case, the frequency offset is best estimated in two passes, first using the STS, and then using the LTS. The IEEE 802.11a standard states that the maximum tolerance for the central frequency is ±20 parts per million (ppm), which corresponds to a maximum possible frequency offset of 200 kHz, when the carrier frequency is 5GHz.

Frequency recovery schemes for OFDM signals can be divided into three categories:

1. Non-data aided algorithms that are based on the spectral characteristics of the received signal.
2. Cyclic prefix based algorithms that use the structure of the signal.
3. Data aided algorithms that are based on known information embedded in the received signal

Non-Data Aided Frequency Synchronizers

The Non-data aided synchronizers can be classified as open loop and closed loop. In an open loop synchronizer, a non-linear element, such as a squaring circuit, is used to generate a frequency component at a harmonic of the carrier frequency. The signal is then filtered to isolate this harmonic and stepped down to the desired carrier frequency. The advantage of these systems is their simplicity and low cost of implementation. However, because of the sensitivity of OFDM signals to frequency offset, open loop synchronizers are generally not practical for OFDM receivers [6].

Closed-loop synchronization uses comparative measurements on the incoming signal and a locally generated signal. Non-data aided algorithms do not need special synchronization blocks, increasing the data throughput and reducing the time needed to achieve synchronization by eliminating the wait for synchronization. For these reasons, these algorithms are well suited for continuous broadcast OFDM signals. Packet-based OFDM signals are also not well suited to this kind of synchronization, since the accuracy is not great enough to ensure orthogonal during the entire packet transmission time [18,19].

Cyclic Prefix Based Frequency Synchronization

Cyclic prefix algorithms are based on an analysis of the sampled received signal before it is passed through the FFT for demodulation. They make use of the redundancy introduced by the inserted guard interval in the OFDM symbol. Since the guard interval is a repetition of the transmitted OFDM symbol over some fraction of the OFDM symbol period, these algorithms simply compare the samples from the data portion of the symbol and the corresponding samples from the guard interval portion of the symbol. If there is a frequency offset of Δf in the receiver frequency, the two values will be different by a factor of $e^{j2\pi\Delta fT}$ where T is the time difference between the two values. This phase difference is proportional to the magnitude of the

frequency-offset error and can then be used as the error signal that drives a voltage-controlled oscillator. The computational complexities of these algorithms are less than the other two categories, therefore providing faster synchronization with lower hardware cost [7,8].

Data-Aided Frequency Synchronizers

Data-aided frequency synchronization provides the best frequency tracking with the widest acquisition range, but at the cost of requiring the use of synchronization blocks. This increases the required overhead and reduces the data throughput[13]. However, for packet-based transmission systems, such as Standard 802.11a, they are required to obtain synchronization quickly before the data information is passed to the receiver. For Standard 802.11a systems, synchronization must occur within the short and long training symbols, which make up the first 16 μs of the packet. The basic algorithm assumes a sequence of repeated training symbols. Similar to the method used in the cyclic prefix algorithms, a comparison is made of the phase difference between adjacent, repeated data symbols. This phase difference is used to generate an error signal that drives a voltage-controlled oscillator.

Coarse Time Synchronization

The function of coarse timing is to find the start of an incoming data packet. In packet switched networks, each packet has a preamble. This section covers each of the coarse time synchronization possibilities .The two algorithms under consideration are the "Basic Auto-Correlation Difference method" and "Auto-Correlation Sum" method [9]

Basic Auto-Correlation Difference Method

This method relies on calculating R (d), and then calculating another auto-correlation sequence, this time with a sample separation of 2L

$$R_2(d) = \sum_{m=o}^{L-1} \left(r_{d+m}^{*} r_{d+m+2L} \right) \quad \text{----------3.8}$$

The difference between these two sequences is then calculated as:

$$R_{diff}(d) = R(d) - R_2(d) \quad \text{----------3.9}$$

In this method, the 16-sample R (d) calculation is reused, and a 32-sample auto-correlator is introduced. The outputs from these two correlators are subtracted, and the output of this subtractor is fed to a peak detector. The location of this peak is taken to be the timing offset point. This difference sequence typically has a triangular peak during the LTS guard interval, and the index, ddiffmax of this peak can be used to calculate the timing offset. This algorithm promises improved performance, and has relatively low hardware complexity.

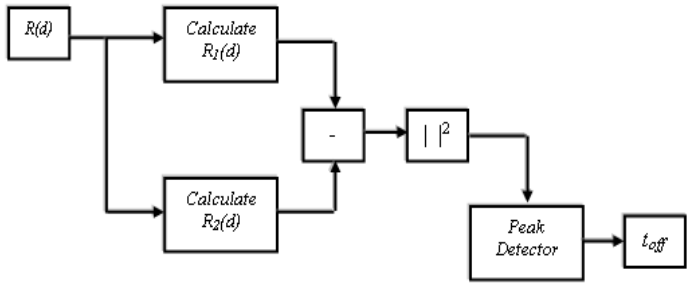

Fig 2: Block Diagram for the Auto-Correlation Difference Algorithm

Auto-Correlation Sum Method

This method calculates the sum of the incoming sequence delayed by L, and the same sequence delayed by 2L, and this sum is correlated with the undelayed sequence. In this method, the calculation of R (d) is reused, with the addition of a delay element, which delays the incoming samples by 32 clock cycles

$$R_3(d) = \sum_{m=0}^{L-1} \left(r_{d+m} \times (r_{d+m+L} + r_{d+m+2L}) \right) \quad \text{----------3.10}$$

Once again, a detector can be designed to determine the index, dsumdrop, at which R3 (d) drops off to half of its peak value.

Fine Time Synchronization

Timing synchronization has two aspects. The first is synchronization with the OFDM symbols, and the second is the synchronization with the data symbols within each OFDM symbol. The synchronization of the OFDM symbols requires more than just matching the symbol timing of the transmitter with the receiver. After calculating a coarse timing offset value using the STS, fine time synchronization can be calculated using the LTS. This involves adding more hardware to the circuit to implement a cross-correlation calculator. It also involves selecting an appropriate detection metric[10,11,12].

Cross-Correlation Calculation

Instead of correlating the incoming sequence with delayed signal samples, it is possible to correlate the incoming sequence with the original preamble sample values. This approach is referred to as cross-correlation, and the calculation is given as:

$$\Lambda(d) = \sum_{m=0}^{L-1} c_m^{*} r_{d+m} \quad \text{----------4.1}$$

The c*m terms are the complex conjugates of the preamble sample values, L is symbol length, r_d is the received sequence and m is an integer.

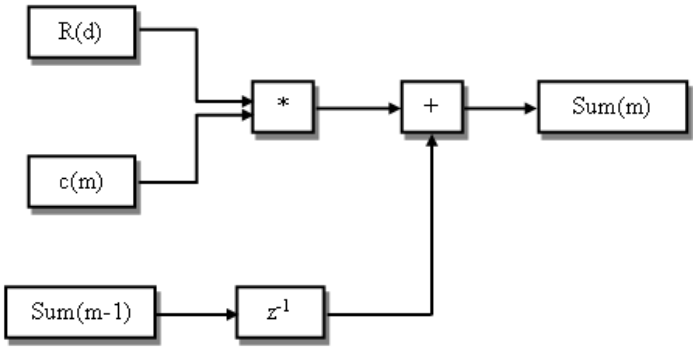

Fig 3: Cross Correlator

In the case where the LTS is used for crosscorrelation, L = 64, and the c*m terms are taken from the original LTS. The crosscorrelation algorithm uses the LTS, and several detectors, which can be used, for determining the timing point is compared. The first of these detectors simply finds the maximum value of $\Lambda(d)$.

$$d_{xc\max} = \arg\max_{d}(|\Lambda(d)|) \quad \text{----------4.2}$$

The second detector adds the absolute values of N successive cross-correlation results, and attempts to maximize the sum:

$$d_{xc\max} = \arg\max_{d}\left(\sum_{p=0}^{N-1} |\Lambda(d+p)| \right) \quad \text{----------4.3}$$

Finally, a third detector looks to find the first instance at which $\Lambda(d)$ exceeds a chosen threshold, th, where th is a percentage of the observed maximum value. The circuitry required for this cross-correlator is composed of multipliers and adders.

Quantized Cross-Correlator

In the quantized version of the cross-correlator the implementation of the multiply accumulate circuitry is modified to reduce hardware complexity. Implementing this quantized cross-correlation involved replacing the multipliers in the original cross-correlation circuit with bit-shifters. It also involved taking the constant values that are used in the cross-correlator and replacing them with quantized values, all of which are powers of 2. Once again, in the case of perfect time synchronization, the sample value at which the cross-correlation would be maximized would be sample 85.

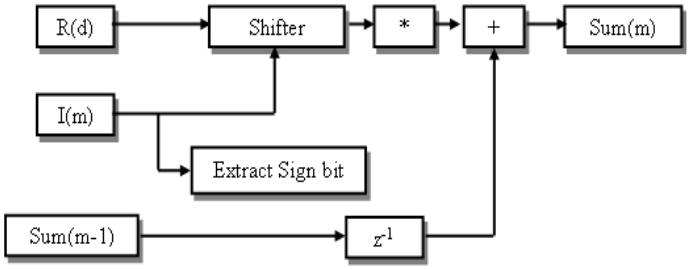

Fig 5: Diagram of the Multiply-Accumulate Circuit Used in the Quantized Version of the Cross-Correlator

Schmidl and Cox Method

In Schmidl and Cox method [5], timing synchronization is achieved by using a training sequence whose first half is equal to its second half in the time domain. The basic idea behind the technique is that the symbol timing errors will have little effect on the signal itself as long as the timing estimate is in the CP. The two halves of the training sequence are made identical by transmitting a PN sequence (Barker code generator) on the even frequencies while zeros are sent on the odd frequencies [1]. The algorithm defined in [3] has three steps, based on the equation 3.1,3.2,3.3.:In equation, the algorithm has a window length of *N*, which is also the number of sub-carriers. The starting point is the value of *n*, which maximizes*M* (*d*). In fact, from the definition, *P* (*d*) expresses the cross-correlation between the two halves of the window; in above Equation, *R*(*d*) represents the auto-correlation of the second half. When the starting point of the window reaches the start of the training symbol with the CP, the values of *P* (*d*) and *R* (*d*) should be equal giving the maximum value for the timing metric. There are two methods to determine the symbol timing. The first one is just to find the maximum of the metric. The second one is to find the maximum, and the points to the left and right that is 90% of the maximum and then compute the average of these two 90% points to find the symbol-timing estimate or symbol/frame timing is found by searching for a symbol in which the first half is identical to the second half in the time domain. Then the carrier frequency offset is partially corrected, and a correlation with a second symbol is performed to find the carrier frequency offset [5].

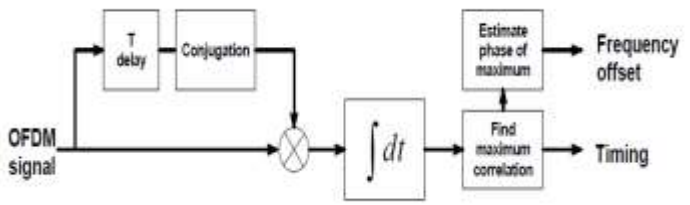

Fig 6 Basic correlation process

Final Implementation

The final implementation of the synchronizer includes the packet detection, frequency offset estimator and timing offset estimator. On the basis of performance capability and low hardware complexity, the Basic Auto-Correlator is preferred for coarse time synchronization and due to low incremental cost the quantized fine time estimator is preferred. The additional size required by the quantized estimator is not a big issue considering the large size of the FPGA being used[17].

Top module of synchronizer consists of quantization, Match filtering and Output block.

Quantization block is used to reduce hardware complexity, which involves replacing the multipliers in original cross correlation circuit with bit shift registers. Basically quantization is the process of mapping a large set of input values to a smaller set, which yields two-power spectrum without loss of information. Let us taking the 8-bit input values 8h'50 and 8h'80 real and imaginary respectively. Then the quantization block is generating sets of values 16h'0000, 16h'8000, 16h'C000, and 16h'E000 and so on real as well as imaginary values.

The matched filter is the second block in our coding. As by definition in communication system, which sends binary messages from the transmitter to the receiver across a noisy channel, a matched filter can be used to detect the transmitted pulses in the noisy received signal. A matched filter is obtained by correlating a known signal, or *template*, with an unknown signal to detect the presence of the template in the unknown signal. This is equivalent to convolving the unknown signal with a conjugated time-reversed version of the template. This section consists correlation unit, magnitude simplified unit and peak finding unit. Correlation unit simplifies the complex multiplication operation to addition operation. We have designed a counter which is set at the value of 16, because the length of the STS is 16 and it is used for enable of correlation is set high, With our input values this unit generates the set of real and imaginary values which are 21 bit long like 21h'025CBC, 21h'0256DA, 21h'02569A and so on real values and 21h'1EBF08, 21h'1EB926, 21h'1ECADA and so on imaginary values.

Magnitude simplified unit replaces the magnitude of complex number with the absolute value i.e. sum of real part and imaginary part. So the output of correlation unit is of 22 bit long and according to the preceding unit the set of outputs are 22h'039DB4, 22h'038C90, 22h'039505 and so on.

The preamble of IEEE 802.11a standard has been designed to help detect the starting edge of the packet. This method, also called the Delay and Correlate Algorithm, takes advantage of the periodicity of the short training symbols at the start of the preamble. The simplest algorithm for finding the start edge of the incoming packet is to measure the received signal energy. When there is no packet being received, the received signal consists only of noise. When the packet starts, the received energy is increased by the signal component, and then the packet can be detected as a change in the received energy level. The packet was set to start at n = 400 and the window length is 22, threshold can be taken within 10 to 35, in order to satisfy the IEEE 802.11a requirements. Once the start of the packet is received, the cross-correlation of the periodic short training symbols causes received signal to jump to the maximum value. This jump gives quite a good estimate of the start of the packet. Peak finding unit is used for finding such of the frame. In this unit According to preamble structure of IEEE802.11a STS counter is set to 0 to 9. On the basis of 22 bit data stream and the received signal during the designated preamble pulse (the 1st and the last, 10th pulse). Once the (conjugate pair) signals are

acquired, the correlation takes place. The phase information of the two signals should yield zero: only the energy level of the correlation result is our concern. Next, the correlation of the 2nd preamble pulse and its counter part, conjugate of the 9th pulse is performed. As long as the correlation is below a pre-defined threshold, the iteration continues until the last pair of preamble pulses (5th and 6th) in the mid section of the short preamble time frame, completes its correlation. The iteration (recursive correlation) will terminate if any signal pair correlation energy level exceeds a predefined threshold. Only then, the receiver announces the detection of signal packet.

Output block consist of a counter, which identifies the data. The value of counter is set according to the IEEE802.11a standards. If counter is between 0-63 the data is input data while for the value between 64-79 the input datas are the cyclic prefix of next signal. So at the output of output block according to the counter we get the recovered data.

The block diagram for the final synchronizer is shown in Figure 7

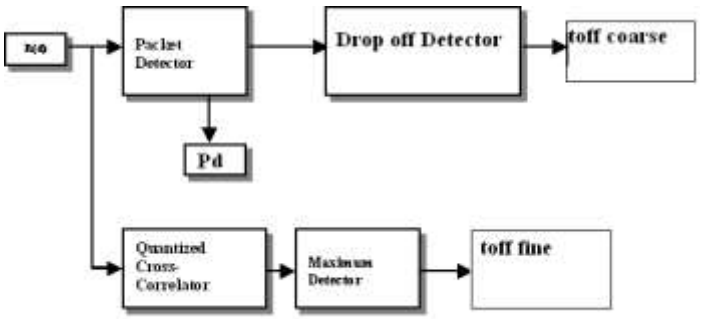

Fig 7: Final synchronizer and algorithm

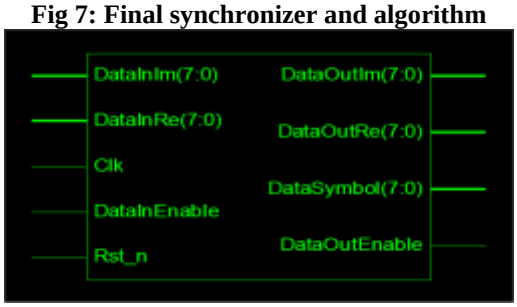

Fig 8: Top module of timing synchronizer

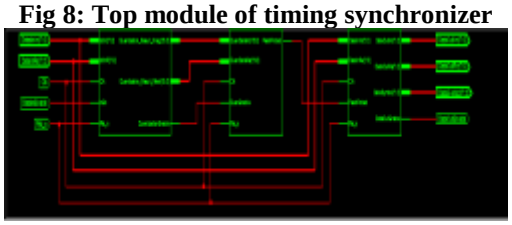

Fig 9 Internal Structure of Synchronizer

Fig 5 design summary of synchronizer

Logic utilization	Used	Available	Utilization
Number of slices	1140	6144	18%
Flip-flops	1137	12288	9%
Number of LUTs	2006	12288	16%
Number of Bonded IOBs	44	240	18%
Number of GCLKs	1	32	3%

Total Hardware Requirements

The hardware requirements for the combination of the packet detector, the frequency offset estimation circuitry, the Basic Auto-Correlator, and the quantized crosscorrelator with the detector are given in fig 9. Minimum period required for synchronizer is 5.885ns and Maximum Frequency is 169.932MHz. Minimum input arrival time before clock is 1.516ns and Maximum output required time after clock is 3.856ns and the total memory usage is around 240 MB.

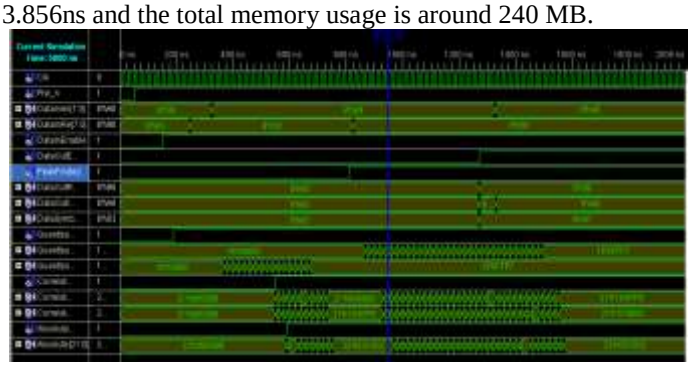

Fig 10 Simulation waveform of synchronizer

Conclusion

After a careful analysis of competing algorithms, it is decided that the best choice for time synchronization to use the Basic Auto-Correlation estimator. It is also decided that the quantized cross correlator, in conjunction with the detector, would be used for fine time synchronization.

References

[1] Rakhi Thakur and kavita khare, "Synchronization and Preamble Concept for Frame Detection in OFDM" 3rd International Conference on Computer Modeling and Simulation (ICCMS 2011) IEEE 978-1-4244-9243-5/11/$26.00 C 2011 V1-46.

[2] Rakhi Thakur and kavita khare, "Synchronization Techniques in OFDM Systems" IRACST International Journal of Computer Networks and Wireless Communications (IJCNWC), ISSN: 2250-3501 Vol.2, No6, December 2012 pp 693-696.

[3] Rakhi Thakur,Kavita Khare et al. "Frame Detection For Synchronization In OFDM" International Journal of Engineering Science and Technology (IJEST) Vol. 3 No. 7 July 2011pp 5955-5957.

[4] Rakhi Thakur,Kavita Khare et al. "Data randomization for synchronization in OFDM system" International Conference on Computing, Communication and Control (ICAC3) January2011 pp28-29.

[5] T.M.Schmidl,D.C.Cox, "Robust Frequency and Timing Synchronization for OFDM" IEEE Transuctions. On communications, dec1997 vol 45, no.12 pp 1613-1621.

[6] H. Minn, M. Zeng and V.K. Bharagava, " On timing Offset Estimation for OFDM Systems", IEEE Communication Letters, July2000, vol 4 no.7, pp. 242-244.

[7] P. H. Moose, "A technique for orthogonal frequency division multiplexing frequency offset correction," IEEE Transactions on Communications, October 1994 vol. 42.

[8] Byungjoon Park, P., Hyunsoo, C., Changeon, K., and Daesik, H "A novel timing estimation method for OFDM systems" IEEE Global Telecommunications Conference, 2002 vol. 1pp. 269-272.

[9] Seung Duk choi jung Min Choi and Jae Hong Lee "An initial timing offset estimation method for OFDM Systems in rayleigh fading channel"2006 IEEE.

[10] Yasamin Mostofi, Donald C. Cox "Analysis of the Effect of Timing Synchronization Errors on Pilotaided OFDM Systems" IEEE Communications Society 2003.

[11] Yasamin Mostofi and Donald C. Cox "Timing Synchronization in High Mobility OFDM system IEEE communication society 2004 pp 2402-2406".

[12] A. I. Bo, G. E. Jian-hua, and Wang Yong, "Symbol Synchronization Technique in COFDM Systems" IEEE Transactions On Broadcasting, Vol. 50, NO. 1, March 2004.
[13] Ch. Nanda Kishore1 and V. Umapathi Reddy"A Frame Synchronization and Frequency Offset Estimation Algorithm for OFDM System and its Analysis"Hindawi Publishing Corporation EURASIP Journal on Wireless Communications and Networking Volume 2006, Article ID 57018, Pages 1–16 DOI 10.1155/WCN/2006/57018.
[14] Ang Ken Li,YewKuan Min ,varun Jeoti "Astudy of frame finding timing synchronization for Wimax applications, international conference on intelligent and advanced systems 2007,IEEE
[15] Yasamin Mostofi and Donald C. Cox "A Robust Timing Synchronization Design in OFDM Systems–Part I: Low-Mobility Cases" IEEE Transactions On Wireless Communications, Vol. 6, No. 11, November 2007.
[16] David Perels, Christoph studer, and W.fichtner "Implementation of low complexity Frame-Start detection algorithm for MIMO system,2007 IEEE.
[17] A. Omri and R. Bouallegue "New Transmission Scheme For MIMO-OFDM System" International Journal of Next-Generation Networks (IJNGN) Vol.3, No.1, March 2011 pp 11-19.
[18] Julien Lamoureux and Wayne Luk, "An Overview of Low-Power Techniques for Field-Programmable Gate Arrays", NASA/ESA Conference on Adaptive Hardware and Systems IEEE 2008 pp 338- 345
[19] Ahmad R. S. Bahai and Burton R. Saltzberg the technical handbook "Multi Carrier Digital Communication theory and application of OFDM.
[20] Juha Heiskala and john Terry A theoretical and practical Handbook "OFDM Wireless LANs".

Rakhi Thakur completed her graduation in Electronics and Tele-communication in 2002, and post graduation in Microwave Engineering in 2005 from R.G.P.V. University. She is a research scholar in MANIT, Bhopal. Her research interests are VLSI and Embedded System for Mixed applications. Earlier she was HOD of EC department in SRIST, Jabalpur but since April 2010 she is in Govt. Polytechnic College Jabalpur.