Software Defined Radio Implementation of LTE Transmitter Physical Layer

Sara M. Hassan
Modern Academy
Cairo, Egypt

A. Zekry
Ain Shams University
Cairo, Egypt

M. A. Bayomy
Modern Academy
Cairo, Egypt

G. Gomah
National Institute for Standards
Giza, Egypt

ABSTRACT
Long Term Evolution (LTE) is an advanced standard of the mobile communication systems. LTE has been developed by the 3rd Generation Partnership Project (3GPP). The new features exhibited by LTE is a direct impact of applying new modulation and coding techniques such as the Orthogonal Frequency Division Multiplexing (OFDM) for the Downlink and the Single Carrier Frequency Division Multiple Access (SC-FDMA) for the Uplink as well as turbo coding. This paper presents a Field Programmable Gate Array (FPGA) design and implementation of the transmitter of the LTE downlink physical layer according to releases 8 and 9 on Virtex 6 XC6VLX240T FPGA kit using Xilinx® ISE® Design Suite version 12.1.

General Terms
SDR**,** LTE, 4G, 3GPP, OFDM, Transmitter , 2G ,3G, LTE downlink physical layer, release 8, release 9 *Xilinx Design Suite, virtex 6 XC6VLX240T FPGA.*

1. INTRODUCTION
The Long Term Evolution (LTE) is an advanced standard for wireless voice and data communication. It is considered a development for the 2G and 3G standards. The main advantages of the LTE, also known as 4G, over the 2G and 3G systems are utilizing a higher peak data rates in both Uplink of 50 Mbps and Downlink of 100 Mbps, scalable bandwidth, and improved spectral efficiency of 5 bps/Hz for Downlink and 2.5 bps/Hz for Uplink. The Physical layer of LTE system applies advanced modulation and coding technologies that are new to cellular applications. OFDM is used in the Downlink while the Single Carrier Frequency Division Multiple Access (SC-FDMA) is used in the Uplink. Also, Multiple Input Multiple Output (MIMO) data transmission and scalable bandwidth scheme up to 20 MHz are utilized in LTE [1-5].

The continued increase in the number of users of mobile communications all over the world, motivated researchers to search for a unified wireless platform. This will enable the mobile users to conduct business and exchange data easily while moving elsewhere in the world. The new features of LTE made it the promising platform intended for the advancements in mobile communications [6]. In the last few years many researchers are concerned with the development of implementation methods and techniques for the building blocks of the LTE physical layer. So, an extensive research work is directed to the implementation of the building blocks of this system using different platforms [7-10].

This paper presents the FPGA design and implementation of the transmitter of LTE downlink physical layer according to release 8 and 9 using Xilinx package version 12.1. All the stages of transmitter are modeled and implemented on Virtex 6 XC6VLX240T FPGA kit.

2. TRANSIMITTER IMPLEMENTATION
The LTE implementation in this paper is built according to specifications of LTE in release 9 with FDD frame structure. The implemented system parameters are as follows [11]:

- Carrier bandwidth: 3 GHz.
- The number of sub-channels: 15.
- Number of sub-carrier:180.
- The IFFT size: 256.
- The turbo encoder/decoder rate: 1/3.
- The data input for one OFDM symbol: 96 bits.

Fig 1 shows the block diagram of the implemented transmitter. The transmitter consists of the blocks: the cyclic redundancy check CRC, the segmentation block, the turbo encoder, the rate matching block, the scrambler, the mapper, the inverse Fourier transform block IFFT, and the cyclic prefix insertion block. The first operation in the transmitter is the Cyclic Redundancy Check (CRC); which add redundancy bits for error detection at the receiver. Then the Segmentation block split up the transport block into multiple segments for the LTE turbo encoder limitation. Rate Matching and Scrambler serves for more security to transmitted data. The OFDM comprise the mapping block which maps bits on QAM symbols, the IFFT block which converts the QAM symbols from frequency domain to time domain, and Cyclic Prefix (CP) insertion for combating the Inter Symbol Interference (ISI) and Inter Carrier Interference (ICI).

In the following sections, we will show the VHDL model and FPGA implementation of the different building blocks of this transmitter.

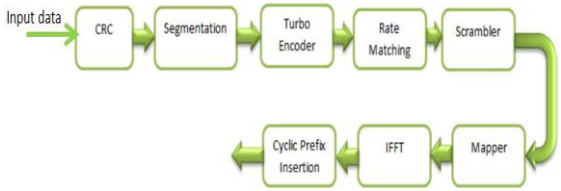

Fig 1: The block diagram of physical layer of the LTE uplink transmitter [11].

2.1 CYCLIC REDUNDANCY CHECK AND SEGMENTATION

There are two types of the CRC blocks in the transmitter. The first one is the first block in the transmitter and it is an independent block. The function of this block is to add 24 redundancy bits at the start of each transport block for sake of detecting errors in the whole transport block. According to the specification, the polynomial used in this block is called "g_{CRC24A}" and it follows the equation [12]:

$$g_{CRC24A}(D) = [D^{24} + D^{23} + D^{18} + D^{17} + D^{14} + D^{11} + D^{10} + D^{7} + D^{6} + D^{5} + D^{4} + D^{3} + D + 1].$$

Where D^n is the location that will be occupied by "1".

The transport block data is modulo 2 divided by this generator polynomial of 24 bits sequence and the remainder is the CRC bit sequence that will be added to the start of the transport block. In case of a transport block has a length greater than 6144 bits, it will be segmented and another CRC bits will be added to each segment. This block is modeled using the hardware description language VHDL and implemented on Xilinix Vertix 6 FPGA chip. The implementation is validated by assuming an input data sequence of 25 ones, then the CRC output data must be '01111001101100110000010 0'. The results of implemented CRC block testing are shown in Fig 2. By inspecting the input and the output sequences in Fig 2, it is noticed they agree fully with the planned test bench results

Fig 2: Testing results of the CRC block.

Fig 2 shows the number of input bits to CRC block: 25 bits, CRC bits with 24 bits as length and the number of output bits: 49 (input + CRC bits).

To enable the Turbo Encoder from handling the data correctly, if the length of the input transport block is greater than 6144, then the transport is split into multiple segments. In this case, another CRC code is added to detect the error in every segment. The CRC inside the segmented block is calculated according to a different polynomial than that used previously in the first CRC block. The polynomial used is called"g_{CRC24B}" and follows the equation [12]:

$$g_{CRC24B}(D) = [D^{24} + D^{23} + D^{6} + D^{5} + D + 1].$$

The output of the first CRC block, which take 49 bits, is utilized an input to the segmentation block. The 49 bits are segmented into 3 segments. The segmentation process is constructed according to the standards [12] of release 9 section (5.1.2) and the length of each segment is according to table (5.1.3-3).This block is modeled using the hardware description language VHDL and implemented on Xilinix Vertix 6 FPGA chip. A test bench is developed to validate the operation of this block using Matlab. With the input sequence from the previous stage, the three segments must be (10101010...101011), (10101010…1101011), and (000010101…110111). The test results of implemented segmentation block are shown in Fig 3. By inspecting the output sequences, a full agreement is noticed with test bench results.

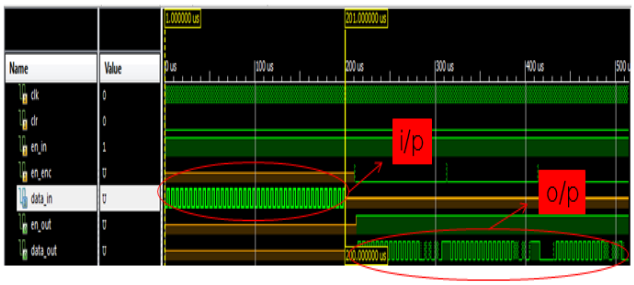

Fig 3: The testing results of the segmentation block.

2.2 turbo encoder

The third block in the LTE transmitter is the Turbo Encoder. The scheme used in the Turbo Encoder is the Parallel Concatenated Convolutional Code (PCCC). As shown in Fig 4 this scheme consists of two 8-state encoders and one turbo code internal interleaver. The coding rate of this turbo encoder is 1/3. The input to the Turbo Encoder is the output of the Segmentation block; which take 100 bits. The three outputs of the Encoder are the systematic ($d_K^{(0)}$) and two sequences of parity bits ($d_K^{(1)}$, $d_K^{(2)}$). The length of each one of the three outputs is 100 bits. The systematic ($d_K^{(0)}$) is the same as the input to Turbo encoder c_K, the two parity sequences ($d_K^{(1)}$, $d_K^{(2)}$) are given by the next equations. The turbo code internal interleaver is constructed according to the standards [11] of release 9 section (5.1.3.2.3).

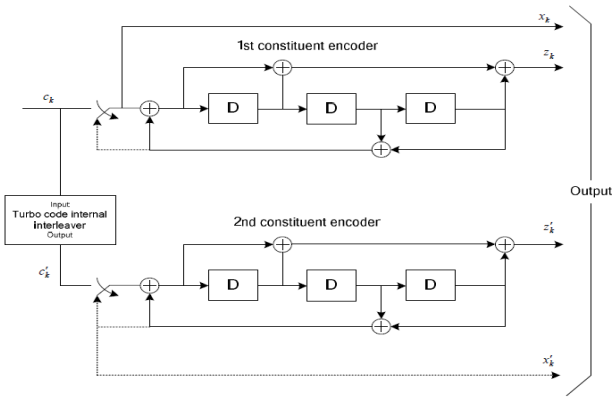

Fig 4: Structure of rate 1/3 turbo encoder [11].

The transmitted bits for trellis termination shall then be [11]:

$$d_K^{(0)} = x_K,\ d_{K+1}^{(0)} = z_{K+1},\ d_{K+2}^{(0)} = x'_K,\ d_{K+3}^{(0)} = z'_{K+1}$$

$$d_K^{(1)} = z_K,\ d_{K+1}^{(1)} = x_{K+2},\ d_{K+2}^{(1)} = z'_K,\ d_{K+3}^{(1)} = x'_{K+2}$$

$$d_K^{(2)} = x_{K+1},\ d_{K+1}^{(2)} = z_{K+2},\ d_{K+2}^{(2)} = x'_{K+1},$$
$$d_{K+3}^{(2)} = z'_{K+2},$$

Where K is the number of input bits.

The turbo encoder is automatically generated by the Xilinx core. A test bench is developed to validate the operation of this block using Matlab. The input and output sequences of the correctly operating turbo encoder must be as follows:

Input:
10101010101010…….……..….10.

Systematic:
10101010101010…….……....10.

Parity
(1):1011001010110110000101110 10…….…..…....….10.

Parity
(2):1101011010110101101011010110 11……01.

The testing results of the implemented turbo encoder generated by Xilinix core is shown in Fig 5. Comparing the output testing results with that of the test bench according to the Matlab, full agreement is found.

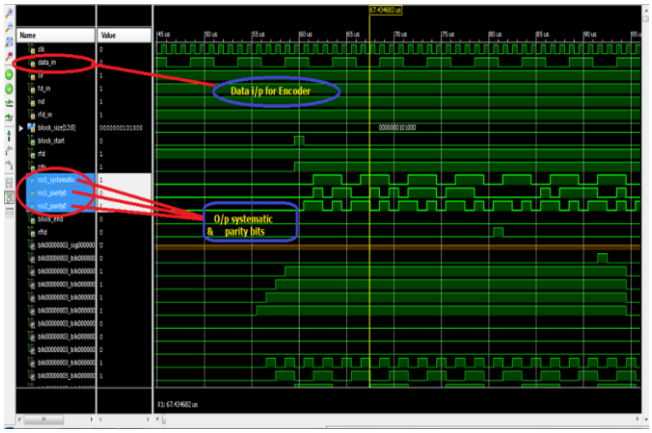

Fig 5: The testing results of the turbo encoder.

2.3 rate matching

The next stage is the Rate Matching block. The function of this block is to add more security for the transmitted data and more immunity against the channel errors. Fig 6 shows the schematic diagram of the rate matching block.

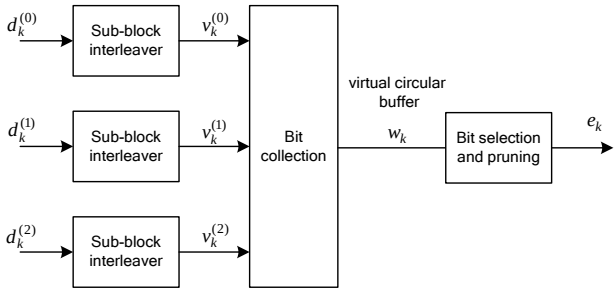

Fig 6: Rate matching block for turbo coded transport channels [12].

The operation of the Sub-block interleaver is to reorder the positions of the input bit stream according to Table 1.

Table 1: Inter-column permutation pattern for sub-block interleaver [12].

Number of columns $C_{subblock}^{CC}$	Inter-column permutation pattern $< P(0), P(1), ..., P(C_{subblock}^{CC} - 1) >$
32	< 1, 17, 9, 25, 5, 21, 13, 29, 3, 19, 11, 27, 7, 23, 15, 31, 0, 16, 8, 24, 4, 20, 12, 28, 2, 18, 10, 26, 6, 22, 14, 30 >

The function of the bit collection is to group the three sequences into one output sequence by taking all the bits of the $v_k(0)$ followed by a bit alternation between $v_k(1)$ and $v_k(2)$. The bit collection operation is shown in Fig 7.

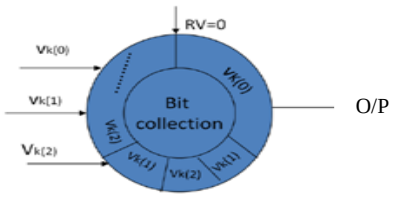

Fig 7: the transmission bit collection and selection operation.

This block is modeled using the hardware description language VHDL and implemented on Xilinix Vertix 6 FPGA chip. A test bench is developed to validate the operation of this block using Matlab. Fig 8a and 8b show the testing results of implemented Rate Matching block. Fig 8a shows the three input streams before and after the interleaving process while Fig 8b shows the output stream of the Rate Matching block. The output bit sequence is identical to that obtained by Matlab, which verifies the correctness of the design of this block.

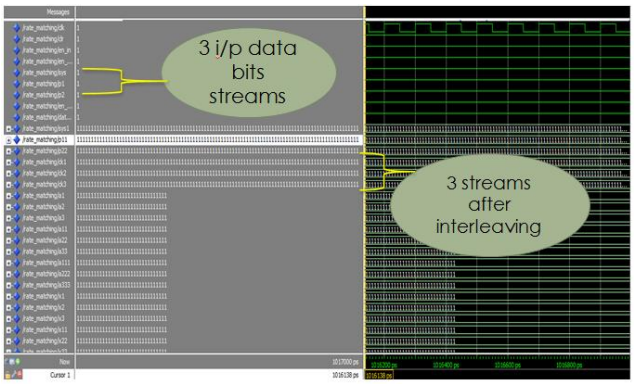

Fig 8a: Testing results of the Rate matching operation.

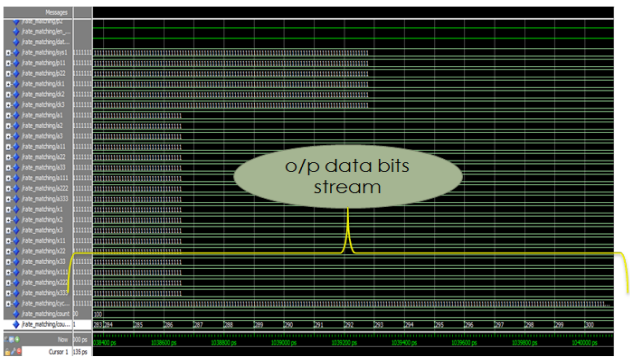

Fig 8b: Testing operation of Rate matching operation.

2.4 Scrambler

The next stage is the Scrambler. There are two main functions for the Scrambler; which are increasing the system security and prevent long sequences of ones ore zeros in the transmitted data to facilitate the clock regeneration in the receiver. The two functions are done through using a Pseudo Random Sequence Generator (PRSG) having a length of 31 bits that is Xored with the data to produce the Scrambler output.

Following the same procedure with the previous building blocks, The testing results of the implemented scrambler is shown in Fig 9, where "data_in" is the input data to Scrambler, "q_out" is the output from the PRSG, and "data_out" is the output data from Scrambler such that (data_out = data_in Xor q_out). Here also we validated the correct operation of the implemented scrambler.

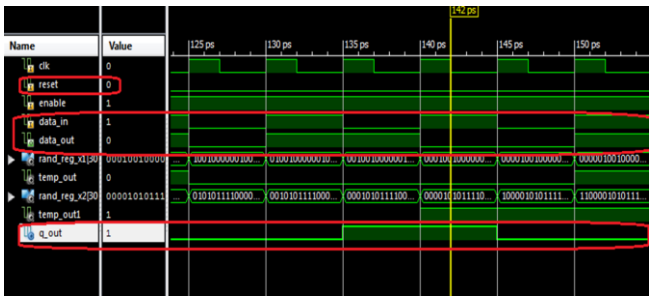

Fig 9: the testing results of the scrambler.

2.5 Mapper

The output data from the Scrambler is the input to the Mapper block, the implementation of the Mapper is carried out by assigning QAM symbols to the input bits according to the constellation diagram. The RTL Schematic of the mapper is shown in Fig 10. Following the same steps of the modeling, implementation and testing as before, the testing output results of the mapper is depicted in Fig 11.Validation showed that the circuit operates correctly.

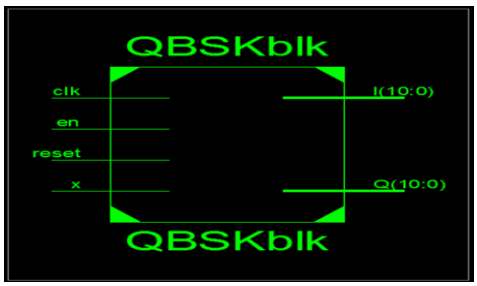

Fig 10: The Mapper RTL Schematic.

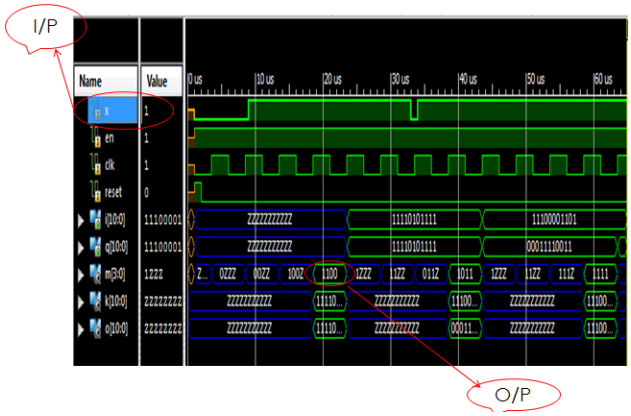

Fig 11: Testing results of the QAM mapper.

2.6 IFFT and Cyclic Prefix

The last two operations in the transmitter increase the bandwidth efficiency through orthogonality by using the Inverse Fast Fourier Transform IFFT and preventing the transmitted symbols from ISI and ICI through the insertion of Cyclic Prefix CP. Both the IFFT block and the CP block with the desired specifications are automatically generated by Xilinx. For the implementation here we use 256 IFFT size (38*2 NULLS + 180 data + pilots). Then the output will be 256 samples for the IFFT. Fig 12a and 12b show the input and output of the implementation of the IFFT core by Xilinx. While Fig 13 shows the implementation of the CP adding by Xilinx according the specifications for short CP, we repeated the first 144 samples at the end of ODFM symbol). As has been done with the previous blocks, the verification of the output signals are carried out by comparing them with that of the Matlab implementation

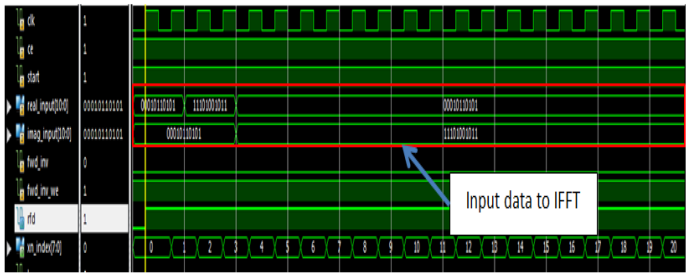

Fig 12a: The input to the IFFT block.

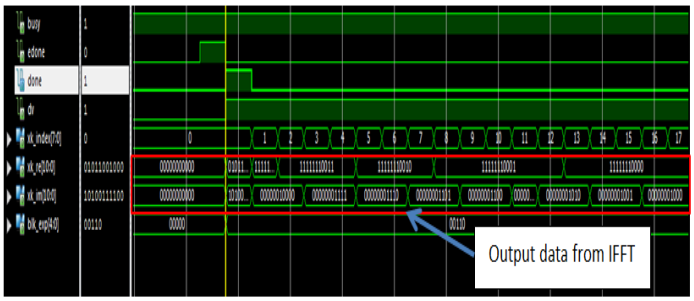

Fig 12b: The output from the IFFT block.

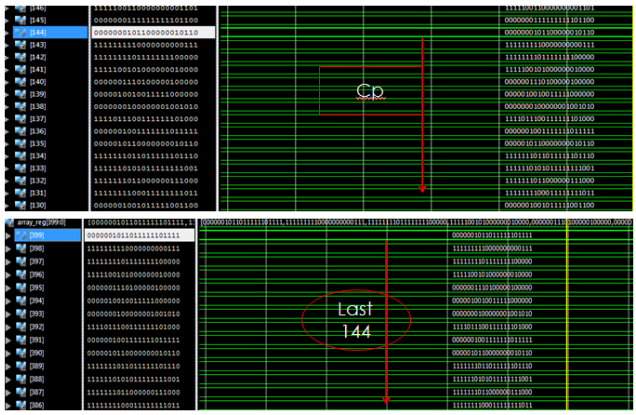

Fig 13: The Cyclic Prefix process.

2.7 Final transmitter implementation

The whole Transmitter implementation is shown in Fig 14 and 15; Fig 14 shows the input to the transmitter. It is two OFDM symbols of 192 bits. Fig 15 shows the output data from the transmitter. It is noticed that the output is identical to that of last IFFT+CP blocks introduced in the previous section as a last building block in the transmitter.

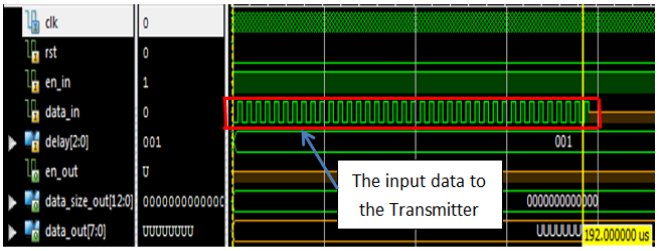

Fig 14: The Transmitter input.

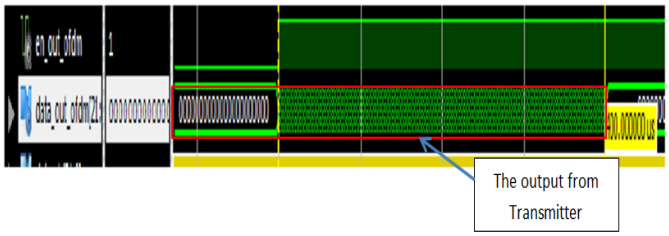

Fig 15: The Transmitter output.

The Transmitter resources utilization table on the Virtex 6 FPGA is shown in table (2). It is clear from the table that the LTE transmitter physical layer utilizes a small fraction of resource blocks from Vittix 6. The chip can utilize the receiver an addition to the transmitter. This will be carried out in the near future.

Table 2: The Transmitter utilization on Virtex 6.

Device Utilization Summary (estimated values)			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Registers	11447	301440	3%
Number of Slice LUTs	12021	150720	7%
Number of fully used LUT-FF pairs	2860	20608	13%
Number of bonded IOBs	29	400	7%
Number of BUFG/BUFGCTRLs	3	32	9%

3. CONCLUSION

In this paper all the stages of the LTE downlink physical lay transmitter are modeled using Xilinx® ISE® Design Suite version 12.1 and implemented on Virtex 6 XC6VLX240T FPGA kit. Applying the Software Defined Radio (SDR) technology in the implementation of this promising standard facilitates applying the anticipated modifications and enhancements that will occur to that standard. It is found that the whole LTE transmitter physical layer consumes a small fraction of the Virtix 6 chip. The chip can accommodate also the the LTE receiver physical layer. This will be done in the near future.

4. REFERENCES

[1] Jim Zyren,Wes McCoy, "Overview of the 3GPP Long Term Evolution Physical Layer", White Paper, freescale semiconductor.

[2] Y. Tsai. G. Zhang. D. Grieco. F. Ozluturk, "Cell Search in 3GPP Long Term Evolution Systems", IEEE Vehicular Technol. Magazine, vol.2, no.2, June 2007, pp.23-29.

[3] Rohde Schwarz, "UMTS Long Term Evolution (LTE) Technology Introduction", Rohde Schwarz, Application Note 1MA111, Sep. 2008.

[4] Motorola, Inc.," Long Term Evolution (LTE): A Technical Overview", Motorola, Inc., Technical White Paper, 2007.

[5]S. Maruyama, S. Ogawa, K. Chiba, "Mobile Terminals toward LTE and Requirements on Device Technologies", IEEE Symposium on VLSI circuit, pp. 2-5, June 2007.

[6]"LTE: The Future of Mobile Broadband Technology", white paper, Verizon wireless .

[7]LTE; Evolved Universal Terrestrial Radio Access (E-UTRA);Physical channels and modulation (3GPP TS 36.211 version 8.3.0 Release 8), ETSI TS 136 211 V8.3.0 (2008-11), Technical Specification.

[8] Sung-won Kim, Kun-yong Kim, "Physical layer verification for 3GPP LTE (FDD)", Agilent Technologies.

[9]Christian Mehlf□uhrer, Martin Wrulich, Josep Colom Ikuno, Dagmar Bosanska, Markus Rupp "SIMULATING THE LONG TERM EVOLUTION PHYSICAL LAYER", 17th European Signal Processing Conference (EUSIPCO 2009).

[10] A.Manikandan, V.Venkataramanan, M. Kavitha, S.Parvathi ," PERFORMANCE ANALYSIS OF LTE PHYSICAL LAYER BASED ON RELEASE 8&9 THROUGH SIMULINK ENVIRONMENT", International Journal of Advanced Technology & Engineering Research (IJATER).

[11] 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Channels and Modulation, Release 9, 3GPP TS 36.211 V9.0.0 (2009-12), Technical Specification.

[12] 3rd Generation Partnership Project Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA);(Multiplexing and channel coding(Release 9), 3GPP TS 36.212 V9.4.0 (2011-09), Technical Specification.