INTERNATIONAL JOURNAL OF TECHNO-ENGINEERING
IJTE

ISSN: 2057-5688

Multicore MIMO-OFDM LTE Optimizing

Y KONDAIAH [1], M. Veera Venkata Ramana [2], G. Mahesah Babu [3]
[1] Associate professor, Assistant Professor, [3] Assistant professor
Department of Electronics and Communication Engineering
Priyadarshini institute of technology & management, guntur

Abstract—The MIMO-OFDM is the prevailing candidate for 4G and 5G broadband wireless communications. This technology is widely adopted in a variety of communication standards like LTE and LTE advanced. It combines multiple inputs and multiple outputs which improves the capacity by transmitting different signals over multiple antennas and orthogonal frequency division multiplexing which divides a radio channel into a large number of closely spaced sub channels to provide more reliable communications at high speed. The most important blocks in LTE 4G are OFDM symbols generation and OFDM symbols reception based on IFFT and FFT. In this paper, we attempt to optimize these blocks to ensure the multicore aspect of MIMO-OFDM LTE.
Key Terms—MIMO-OFDM, LTE, FFT, IFFT, 4G, 5G

I. INTRODUCTION

In recent years, the rapid development of digital signal processing (DSP), telecommunication and digital communication has imposed a need for high speed data transmission. Developers must find solutions that respond to user requirements in terms of latency, power computation, power consumption and cost. In order to satisfy these requirements, Networkon-Chip (NoC) was developed to be a successful solution for programming applications based on multicore DSP platforms [1]. NoC is an emerging interconnect infrastructure to address the scalability limitation of conventional shared bus architecture for many-core system-on-chip (MCSoC) [2]. In this work, we adopt the Long Term Evolution (LTE) which is the standard radio communication with very high speed and low latency [3]. The most important core of LTE is the Orthogonal Frequency Division Multiplexing (OFDM) which is a popular technique of signal modulation presenting a promising prospect to meet the requirement of modern wireless network system [4]. In addition, OFDM provides high bandwidth efficiency, robustness to multipath fading given that it uses cyclic prefix, and simplicity of its implementation by using dual functions Fast Fourier Transform (FFT) and Inverse Fast Fourier Transform (IFFT). The OFDM transmitter and receiver contain, respectively, the IFFT and FFT which are time consuming critical functions [4]. These modules are computationally intensive ones compared to the other blocks in the chain of MIMO-OFDM LTE [5], [6]. Thus, inefficient IFFT and FFT may decrease the overall system performance. For that reason, the implementation of these two blocks should be optimized to achieve a better performance. In this paper, we attempt to optimize these two blocks by proposing a multicore IFFT and a multicore FFT in the OFDM blocks, i.e, symbols generation and symbols reception. The rest of paper is structured as follows. Section 2 presents the design methodology. Section 3 presents the MIMO-OFDM blocks. Section 4 is devoted to the proposed multicore architecture. Section 5 shows the simulation results of a multicore FFT in PhoenixSim. Finally, section 6 concludes the paper and underlines the future works.

II.LITERATURE SURVEY

In order to obtain a multicore architecture of an application, we will use 3 design flows. The first design flow is an alternative which can be used for any type of application. It is a workflow that starts from a Simulink model and a multicore architecture based on a NoC [7]. The Simulink model is automatically translated into an SDF graph. Although, the transformation is successfully completed, the necessary transformation of the graph is computed and the scheduling decisions are made in compile time. In fact, the graph is mapped and scheduled on the architecture. A set of C/C++ programs describing the behavior of the application can be generated on the diverse cores of the DSP platform based on the mapping and scheduling operations results. In our case, we will opt for mathematical approach instead of this workflow. The second design flow is based on OMNET++ and PhoenixSim in order to evaluate the performance of our application MIMO-OFDM LTE. We use a modified version of PhoenixSim which is a physical layer simulator developed in OMNET++ [8]. PhoenixSim is a simulation environment that is suitable for investigating both electronic and photonic NoC and hybrid NoC (photonic and electronic NoC). The performance metrics of interconnection networks can be analyzed both at the system level (latency, throughput, execution time) [9] and physical layer (energy dissipation power both static and dynamic where power modeling ORION is used nearly in every electronic component). Finally, the third design flow is the implementation. We will choose ProNoC [10] which is a prototype platform that generates RTL codes of a complete heterogeneous NoC/Wishbone Bus (WB)-based MCSoC. The GUI of ProNoC (developed with Perl language) provides a fast way to generate a synthesizable RTL code of any complex heterogeneous NoC-based MCSoC platforms. Figure 1 illustrates the functional block diagram of a heterogeneous NoC-based MCSoC which can be generated using ProNoC software which consists of processing tile connected by a low latency wormhole VC-based NoC. Processing tile consists of Intellectual Properties (IPs) group that includes wishbone bus, Random access memory (RAM), timer, external interrupt, Network Interface (NI), General Purpose Input Output (GPIO), interrupt controller, Altera JTAG and aeMB processor [11]. These IPs are connected by a shared WB [12]. However, IPs are not limited to the above-mentioned ones. We can use this tool to add any IP once having its RTL Verilog code. In this case, we add the DSP OpenRISC OR1200 [13] as a new IP by collaborating with its designer in order to have a prototype NoC-DSP to be used later for the implementation phase of our application MIMO-OFDM LTE.

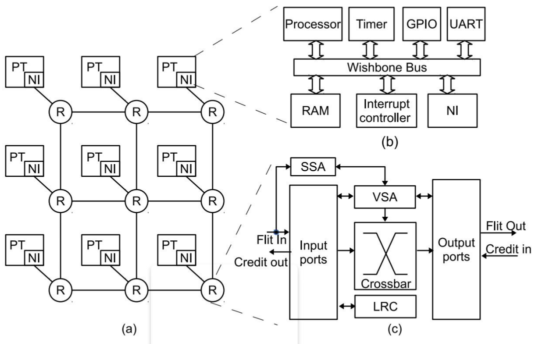

Fig. 1. ProNoC [10] functional block diagram (a) MCSoC, (b) processing tile(PT) (c) a NoC router.

MIMO-OFDM LTE BLOCKS The MIMO-OFDM LTE blocks are illustrated in figure 2 [14]. This section provides a brief description of these blocks.

• Transport blocks Payload bits: The Matlab function genPayload.m creates pseudo-random bits.

• Transport block CRC insertion: The Matlab function CRCgenerator.m inserts CRC parity bits into the transport block.

• ”Segmentation and Code block CRC insertion”, ”Channel encoding”, ”Rate Matching”, ”Code block concatenation”: The Matlab function TbChannelCoding.m realizes these blocks.

• The Matlab function Scramble.m realizes the scrambling operation.

• The Matlab function Modulator.m realizes the modulation QPSK, 16QAM, 64QAM.

• The Matlab function LayerMapper.m maps the data to the layers.

• The Matlab function SpatialMuxPrecoder.m realizes the precoding operation.

• The Matlab function CSRgenerator.m generates Cell Specific Reference (CSR).

• The Matlab function REmapper_mTx.m attributes data and reference signal to the ressource grids.

• OFDM symbols generation: The Matlab function OFDMTx.m realizes the operation of IFFT and the addition of cyclic prefix.

• Channel: The Matlab function MIMOFadingChan.m produces a MIMO Channel with multi-path fading and the Matlab function AWGNChannel.m adds the white Gaussian noise to the channel.

• OFDM symbols reception: The Matlab function OFDMRx .m realizes the operation of FFT and the removing of cyclic prefix.

•The Matlab function REdemapper_mTx.m extracts the data and the reference signal (CSR) from the resource grids.

•The Matlab function ChanEstimate_mTx.m realizes the channel estimation operation.

• The Matlab function ExtChResponse.m extracts the user data from the estimated channel response.

•The Matlab function MIMOReceiver_OpenLoop.m realizes the equalization operation .

• The Matlab function LayerDemapper.m realizes the layer demapping operation.

• The Matlab function DemodulatorSoft.m realizes the demodulation QPSK, 16QAM, 64QAM.

• The Matlab function Descramble.m realizes the descrambling operation.

•”Code block separation”,”Rate dematching”, ”Channel Decoding” and "Code block CRC check and code block concatenation": The Matlab function TbChannelDecoding.m realizes these blocks.

• Transport block CRC check: The Matlab function CRCdetector.m extracts the parity bits of the transport block. In this paper, we are particularly interested in two blocks: OFDM symbols generation and OFDM symbols reception.

A. OFDM symbols generation OFDM symbols generation as shown in figure 3 consists of 4 blocks which are serial to parallel converter, modulator, Inverse Fast Fourier Transform and parallel to serial converter. First, the input symbols are sent serially into the transmitter. Then, these symbols pass to serial to parallel converter. The number of sub-channels specifies the number of bands in which the total spectrum is subdivided. Finally, these symbols are sent to an Inverse Fast Fourier Transform module. IFFT transform a signal from the frequency domain into the time domain in order to preserve the subcarrier orthogonality and the independence of subsequent OFDM symbols, a cyclic prefix (CP) technique is introduced.

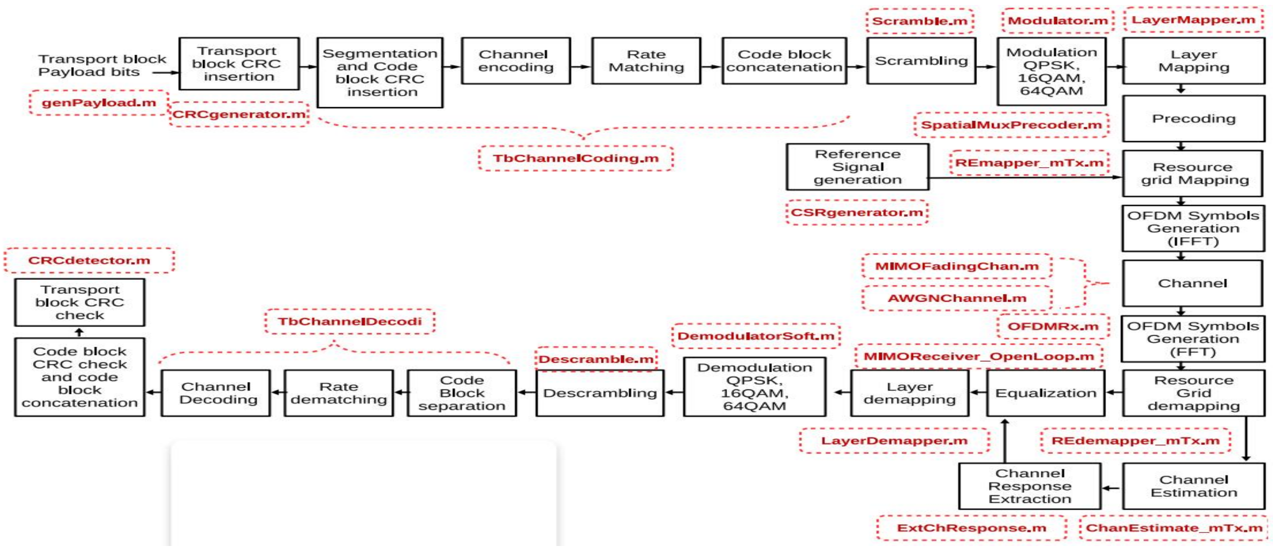

Fig. 2. MIMO-OFDM LTE blocks

B. OFDM symbols reception As presented in Figure 3. OFDM symbols reception is the opposite process of OFDM symbols generation including serial to

ISSN: 2057-5688

parallel converter, Fast Fourier Transform, demodulator, and parallel to serial converter. The incoming signal is produced through a serial to parallel converter. After the CP removal, the received signal is passed through an n point Fast Fourier Transform in order to convert time domain signal to frequency domain.

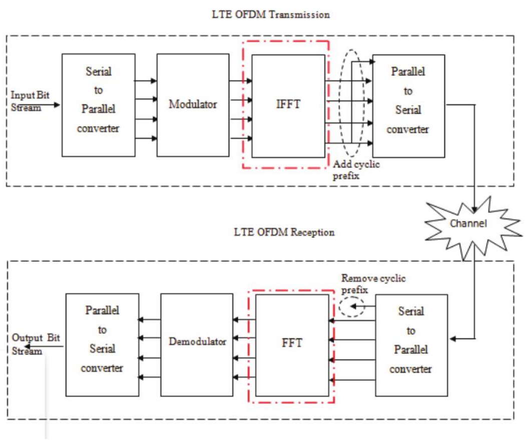

Fig. 3. Block Diagram of OFDM system.

III. PROPOSED SYSTEM

MULTICORE OFDM Our proposed architecture consists of developing a multicore OFDM based on parallel IFFT and parallel FFT [15]. The required FFT or IFFT size in LTE are 128, 256, 512, 1024, 1536, 2048. For a bandwidth of 10 MHz, the size of FFT or IFFT is 1024. The workflow of multicore FFT or IFFT is shown in Figure 4. A. Analysis of multicore FFT/IFFT The size of FFT or IFFT is n, p is the number of PEs in a NoC. Computation: This multicore FFT or IFFT is divided into log2 (n) stages. For stage 0 to stage log2 (n/p)-1: We have p PEs computing in parallel independently, at every stage, each processing element will compute n/p point and every point needs one multiplication and one addition. From stage $\log_2$ (n/p) to stage $\log_2$ (n)-1, we have p/2 PEs compute in parallel after their communication with their corresponding PEs, at every stage, every PE computes 2n/p point and every point needs one multiplication and one addition. The time complexity of our multicore FFT or IFFT is reduced

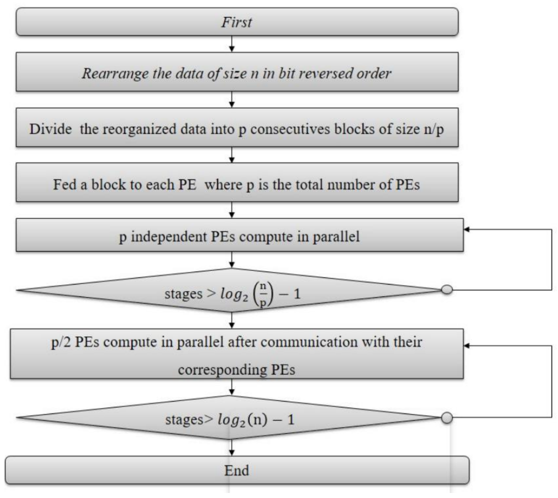
First
Rearrange the data of size n in bit reversed order
Divide the reorganized data into p consecutives blocks of size n/p
Fed a block to each PE where p is the total number of PEs
p independent PEs compute in parallel
stages > $log_2\left(\frac{n}{p}\right) - 1$
p/2 PEs compute in parallel after communication with their corresponding PEs
stages> $log_2(n) - 1$
End

Fig. 4. Flowchart of multicore FFT/IFFT.

to O(nlog$_2$(n/p)) compared to the time complexity of a sequential FFT or IFFT which is O(nlog$_2$(n)). Communication: if p=n: The communication takes place at every stage and there is one data exchange between pairs of PEs at every one of the log$_2$(n) stages. Assume that the interconnection permits a simultaneous exchange, the communication time complexity is O(nlog$_2$(n)).

IV.RESULTS

As a first step, we integrate the MATLAB codes of multicore FFT and multicore IFFT, respectively, in the OFDM symbols generation and OFDM symbols reception blocks. As shown in Figure 5 and Figure 6. The spectra of transmitted and received signals using multicore FFT and IFFT is the same when using sequential FFT and sequential IFFT. A simple example of a multicore FFT with n=128 and p=4 simulated in PhoenixSim can fulfill a preliminary work to achieve our objective. Table I presents the configuration parameters of a multicore FFT.

TABLE I
CONFIGURATION PARAMETERS OF A MULTICORE FFT.

Configuration parameters	
Network configuration	value
Process technology	32nm
Number of tiles	4
Chip area	400 nm
Core frequency	2.5 GHz
Power model	Orion 2.0

Table II presents the evaluation performance of our multicore FFT in terms of simulation time, latency, bandwidth and electronic energy dissipation that is composed of electronic

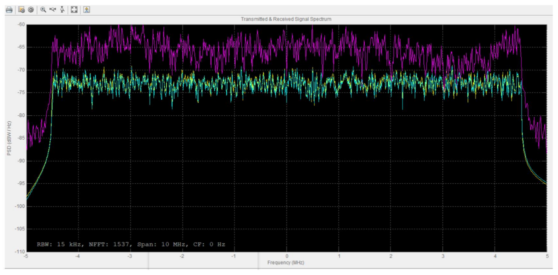

Fig. 5. Spectra of transmitted and received signal using multicore FFT/IFFT.

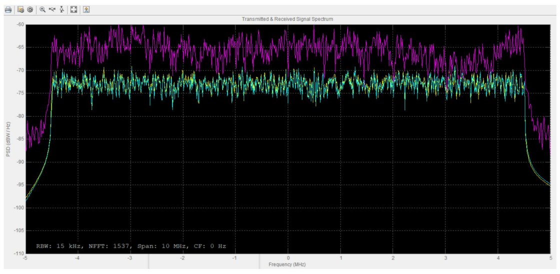

Fig. 6. Spectra of transmitted and received signal using sequential FFT/IFFT.

arbiter, electronic crossbar, electronic input port, and electronic clock where all components are decomposed into static and dynamic energy as shown in Table III.

TABLE II
PERFORMANCE EVALUATION OF A MULTICORE FFT.

Simulation time(s)	0.011375572567
Electronic energy dissipation(J)	0.00787799
Throughput(Gb/s)	0.00072014
Average latency(us)	0.205

TABLE III
DECOMPOSITION OF POWER ELECTRONIC POWER.

Electronic energy	Static	Dynamic
Electronic arbiter	3.53143e-005	2.97545e-011
Electronic clock	0.00120214	0
Electronic crossbar	0.00162934	8.40136e-013
Electronic inport	0.00500954	0
Electronic wire	1.30984e-006	4.61491e-010

We aim to improve the parameters shown in Table II to select the best architecture of our application MIMO-OFDM LTE by exploring and evaluating a variety of heterogeneous architectures designs. The execution time and latency should be reduced in order to run our application in real time. Also, the power dissipation should be reduced and finally, the throughput should be increased.

REFERENCES

[1] M. S. Gaur, V. Laxmi, M. Zwolinski, M. Kumar, N. Gupta et al., “Network-on-chip: Current issues and challenges,” in 19th International Symposium on VLSI Design and Test (VDAT). IEEE, 2015, pp. 1–3.

[2] S. Kumar, A. Jantsch, J.-P. Soininen, M. Forsell, M. Millberg, J. Oberg, K. Tiensyrja, and A. Hemani, “A network on chip architecture and design methodology,” in Proceedings. IEEE Computer Society Annual Symposium on VLSI. IEEE, 2002, pp. 117–124.

[3] A. B. Abdullahi, A. Hammoudeh, and B. E. Udoh, “Performance evaluation of MIMO system using lte downlink physical layer,” in SAI Computing Conference (SAI). IEEE, 2016, pp. 661–668.

[4] S. Nouri, W. Hussain, and J. Nurmi, “Evaluation of a heterogeneous multicore architecture by design and test of an OFDM receiver,” IEEE Transactions on Parallel and Distributed Systems, vol. 28, no. 11, pp. 3171–3187, 2017.

[5] M. Makni, M. Baklouti, S. Niar, M. Biglari-Abhari, and M. Abid, “Heterogeneous multi-core architecture for a 4G communication in high-speed railway,” in Design & Test Symposium (IDT), 2015 10th International. IEEE, 2015, pp. 26–31.

[6] S. Hesham, J. Rettkowski, D. Goehringer, and M. A. A. El Ghany, “Survey on real-time networks-on-chip,” IEEE Transactions on Parallel and Distributed Systems, vol. 28, no. 5, pp. 1500–1517, 2017.

[7] K. Gasmi, A. Rebaya, I. Amari, and S. Hasnaoui, "Workflow for multi-core architecture: From MATLAB/Simulink models to hardware mapping/scheduling," in Information and Digital Technologies (IDT), International Conference on. IEEE, 2017, pp. 142–148.

[8] A. B. Ahmed and A. B. Abdallah, "Hybrid silicon-photonic networkon-chip for future generations of high-performance many-core systems," The Journal of Supercomputing, vol. 71, no. 12, pp. 4446–4475, 2015.

[9] J. Chan, G. Hendry, A. Biberman, K. Bergman, and L. P. Carloni, "Phoenixsim: A simulator for physical-layer analysis of chip-scale photonic interconnection networks," in Proceedings of the Conference on Design, Automation and Test in Europe. European Design and Automation Association, 2010, pp. 691–696.

[10] A. Monemi, J. W. Tang, M. Palesi, and M. N. Marsono, "ProNoC: A low latency network-on-chip based many-core system-on-chip prototyping platform," Microprocessors and Microsystems, vol. 54, pp. 60–74, 2017.

[11] S. T. S. Ngiap. (Online; accessed Dec 2018) Aemb 32-bit microprocessor core. [Online]. Available: http://opencores.org/project,aemb

[12] OpenCores, "Wishbone system-on-chip (soc) interconnection architecture for portable IP cores," https://opencores.org/opencores,wishbone, Online; accessed Dec 2018.

[13] OpenRISC, "OpenRISC 1000 architecture," https://openrisc.io/implementations.html, Online; accessed Dec 2018.

[14] H. Zarrinkoub, Understanding LTE with MATLAB: from mathematical modeling to simulation and prototyping. John Wiley & Sons, 2014.

[15] Z. Cui-xiang, H. Guo-qiang, and H. Ming-He, "Some new parallel fast fourier transform algorithms," in Sixth International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT). IEEE, 2005, pp. 624–628.