IMPLEMENTATION OF LTE UPLINK SYSTEM FOR SDR PLATFORM USING CUDA AND UHD

Yong Jin(HY-MC Research Center, Hanyang Univ., Seoul, South Korea; kimyong84@dsplab.hanyang.ac.kr); Chiyoung Ahn (HY-MC Research Center, Hanyang Univ., Seoul, South Korea; ahncy@dsplab.hanyang.ac.kr); and Seungwon Choi (corresponding author, HY-MC Research Center, Hanyang Univ., Seoul, South Korea; choi@ieee.org)

ABSTRACT

In this paper, we present an implementation of Long Term Evolution (LTE) Uplink (UL) system on a Software Defined Radio (SDR) platform using a conventional Personal Computer (PC), which adopts Graphic Processing Units (GPU) and Universal Software Radio Peripheral2 (USRP2) with URSP Hardware Driver (UHD) for SDR software modem and Radio Frequency (RF) transceiver, respectively. Noting that Central Processing Unit (CPU) control code, i.e., a customized C++ code can access USRP2 via UHD, we have adopted UHD instead of GNU Radio because UHD provides a lot higher degree of flexibility in the design of transceiver chain compared to the GNU Radio. By taking advantage of this benefit, Cognitive Radio (CR) engine have been implemented by using libraries from UHD. Meanwhile, we have implemented the software modem in our system on GPU which is suitable for parallel computing due to its powerful Arithmetic and Logic Units (ALUs). From our experiment tests, we have measured the total processing time for a single frame of both transmit and receive LTE UL data to find that it takes about 5.00ms and 6.79ms for transmit and receive, respectively. It particularly means that the implemented system is capable of real-time processing of all the baseband signal processing algorithms required for LTE UL system.

1. INTRODUCTION

Recently, researches of using Graphic Processing Unit (GPU) and Universal Software Radio Peripheral2 (USRP2) are performed very actively according to active development of various Software Defined Radio (SDR) platforms [1][2]. In order to cope with an extremely high-speed data rate in modern communication systems, the processor for SDR-based signal processing should support the high system throughput and high operation speed. Considering that GPU is suitable for parallel computing due to its powerful Arithmetic Logic Units (ALUs), we have implemented the software modem on GPU in our implemented system. For an efficient use of the ALUs inside a GPU, Compute Unified Device Architecture (CUDA) that is C-based programming environment is utilized for developing a GPU-based SDR platform [3]. Since USRP2, an Radio Frequency (RF) transceiver, can be interconnected with the host Personal Computer (PC) through Gigabit Ethernet while various libraries needed for the host PC to control USRP2 are provided from GNU Radio and Ettus, USRP2 is widely used in most SDR platform developments [2]. In order for USRP2 to interface with PC, a proper driver should be adopted. USRP Hardware Driver (UHD) provided by Ettus has been selected as a driver for interface between USRP2 and PC in our system. UHD has been known to provide a lot more various libraries for controlling USRP2 compared to GNU Radio and, furthermore, new versions are being developed by Ettus [4]. In this paper, we present an implementation of an Long Term Evolution (LTE) Uplink (UL) system, i.e., 3rd Generation Partnership Project (3GPP) Release 9 standard [5][6], using GPU, a typical parallel processor, as its modem chip and USRP2 as its RF transceiver. Implemented LTE UL system is verified in terms of its processing time in RF signal environments. We also present a novel Cognitive Radio (CR) engine which automatically switches its RF frequency in accordance with the change in the transmitting site. The CR engine proposed in this paper has been implemented using UHD on USRP2.

The organization of this paper is as follows. Section 2 introduces an overall structure of implemented system. Section 3 explains how the GPU modem, USRP2 board, and CR engine have been designed. Section 4 summarizes the implementation of the LTE UL system, while Section 5 concludes this paper.

2. STRUCTURE OF IMPLEMENTED SYSTEM

Figure 1 illustrates the structure of implemented LTE UL system consisting of a host PC and USRP2 with an antenna. Waveform application shown in Figure 1 denotes a software modem for decoding LTE UL data. The waveform application shown in Figure 1 has been implemented in such a way that every baseband signal processing required for LTE is performed on GPU using CUDA toolkit which includes various CUDA libraries such as CUFFT, CUBLAS, CURAND, etc. In addition, libraries of GNU Radio for controlling the video stream and libraries of UHD for controlling USRP2 have also been adopted in the waveform application. As shown in Figure 1, the host PC and USRP2 are connected through Gigabit Ethernet, while UHD firmware for the host PC to control USRP2 is prepared in USRP2. The UHD, a driver for accessing USRP2, consists of C++ libraries to be used in the host PC and the UHD firmware to be used in USRP2. Especially, the UHD firmware has been ported on Field Programmable Gate Array (FPGA) in USRP2 as shown in Figure 1.

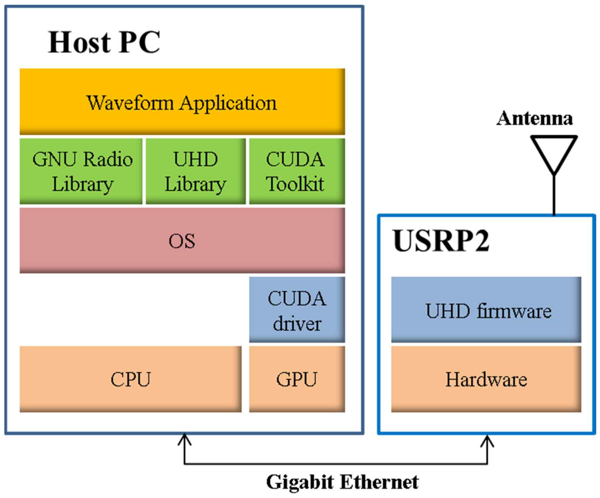


Figure 1. Structure of implemented system

3. IMPLEMENTATION

3.1 GPU modem

As mentioned earlier, baseband signal processing for decoding LTE UL signals is performed on GPU equipped in the host PC. The GPU adopted in our implementation is NVIDIA GTX 295 which is capable of 1788 GFLOPS with 480 CUDA core processor cores [7].

Figure 2 illustrates a block diagram of data format of LTE UL system for designing the implemented GPU modem [5][6]. In the case of transmitter, as Central Processing Unit (CPU) transfers the video stream data to GPU, GPU performs encoding of LTE UL data in accordance with 3GPP Release 9 standard. After the encoding processing is completed, the results are stored in CPU buffer, which is being used for transferring the transmit data to USRP2 via Ethernet.

In the case of receiver, similarly to the case of transmitter, the received data stored in CPU buffer are first transferred into GPU. The received data stored in the CPU buffer are the ones received and frequency-down-converted by USRP2. The transmitted video stream data can be retrieved, as soon as GPU completes decoding of the received data fed from CPU.

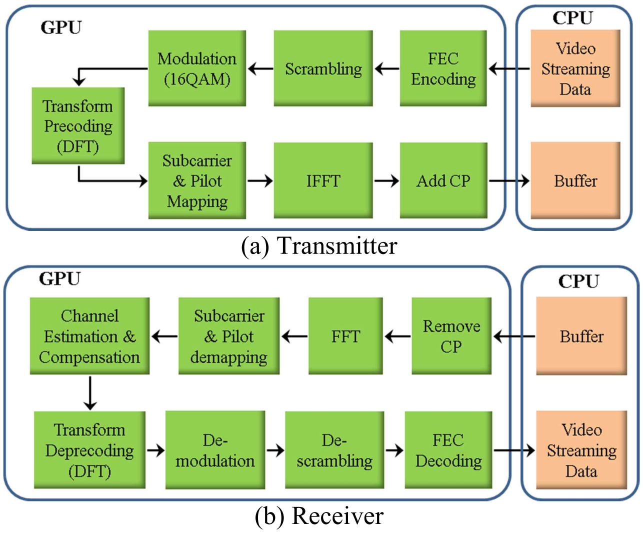


Figure 2. Data format of LTE UL system

3.2 Signal Processing on USRP2 Board

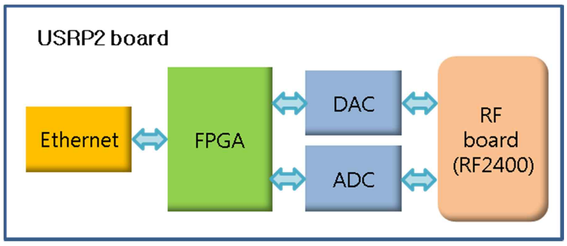


Figure 3. Functionalities of USRP2 Board

Figure 3 illustrates a block diagram of signal processing in USRP2 board in our implemented system. As shown in Figure 3, RF transceiver used in the implemented system consists of USRP2 board and RF board (RFX2400) equipped inside the USRP2 board. The RF board performs the frequency-up or frequency-down conversion of the signal to be transmitted or received, respectively.

Signal processing mechanism in the USRP2 board for signal transmission can be explained as follows. Firstly, signals to be transmitted that have been transferred by the host PC through the Gigabit Ethernet are stored in the internal buffer of FPGA. The transmit data include not only the data, i.e., video stream data in our implementation, to be transmitted but also control signals such as transmit frequency band, transmit signal power, etc, which are transferred from CPU for USRP2 to process the transmit data properly. Then, after the FPGA completes decoding of the control signals, transmission of the data becomes ready by setting up the USRP2 peripherals with the corresponding parameters, i.e., transmit frequency band, transmit signal power, etc, mentioned above. Finally, transmit data stored at the FPGA buffer are converted into corresponding analog signal through Digital to Analog Converter (DAC) and transferred into the RF board, i.e., RFX2400. Receiving mechanism is performed in the exactly reverse order of the transmission procedure explained above.

3.3 CR Engine using UHD

In this section we introduce a CR engine that has been developed using the library provided from UHD [8]. The CR engine provides a functionality of converting frequency band from current one to a desired one in accordance with the result of analysis of received signal quality during run-time. As explained in Section 3.2, the transmit or receive frequency band of USRP2 is determined by the control signal transmitted from the host PC.

The CR engine implemented in our system first transmits the control signal to USRP2 for USRP2 to be able to receive the signal in a default frequency band. If the signal is not captured in that frequency band during a certain period set up a priori, CR engine transmits a control signal once again for the system to change its receive frequency band.

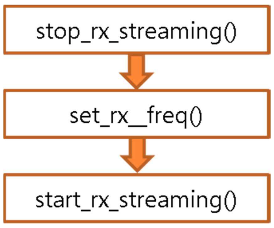


Figure 4. Procedure of receive frequency change

Figure 4 illustrates the procedure of converting the receive frequency of USRP2. The conversion of receive frequency band into a desired one is achieved after terminating the data streaming into the host PC. Finally, the data streaming is continued using the new frequency band.

4. SIMULATION RESULTS

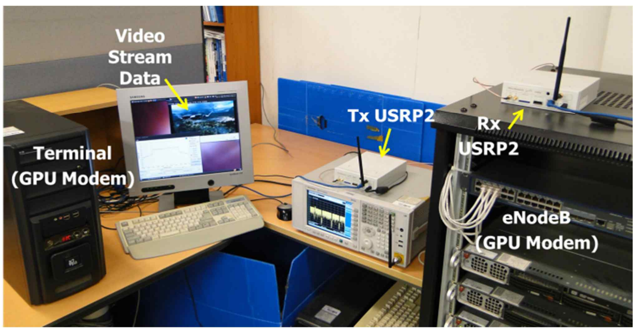
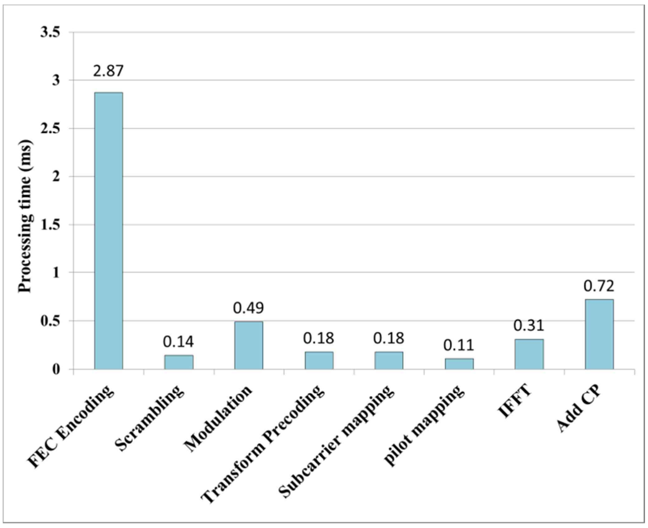
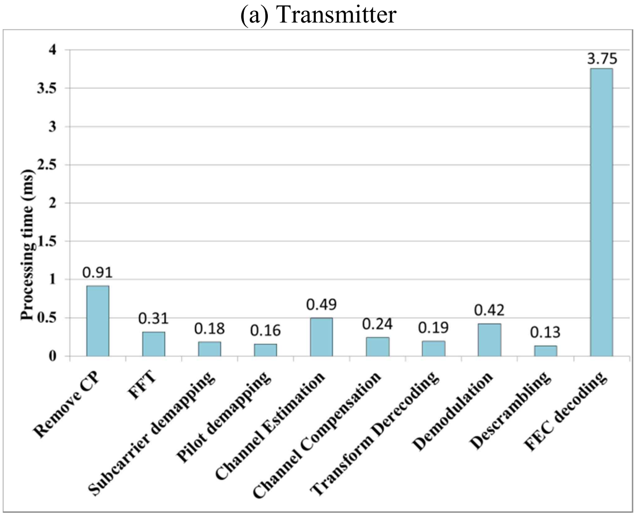


Figure 5. LTE UL system

Figure 5 illustrates the overall environment of our LTE UL experiments consisting of transmitting terminal and receiving eNodeB. As mentioned earlier, our system adopts USRP2 as its RF transceiver, which is interconnected with the host PC through Gigabit Ethernet. Baseband signal processing for decoding the received signal is performed in GPU equipped in the host PC, while CPU controls USRP2 through UHD and replays the received video stream data. The maximum throughput of our implemented system is 8Mbps for the video stream data transmission and reception.



(a) Transmitter



(b) Receiver

Figure 6. Processing time of 1 LTE frame

Figure 6 illustrates the processing time of each signal processing block that is realized in GPU. Each signal processing block implemented in GPU has been designed in such a way that each block is processed using as many as possible number of threads. The processing time shown in Figure 6 has been measured using the CUDA Profiler provided by NVIDIA.

As shown in Figure 6(a), Forward Error Correction (FEC) encoding took the largest processing time, i.e., 2.87ms, among all the transmit operations in LTE UL. In order to encode the entire LTE frame, it took about 5.00ms. From Figure 6(b), the total processing time to decode a single LTE frame is about 6.79ms. Similarly to the case of transmit, FEC decoder took the largest processing time of all the receive operations in LTE UL. The reason is that, the intrinsic properties of FEC encoder and decoder bring a disadvantage in parallel processing using GPU.

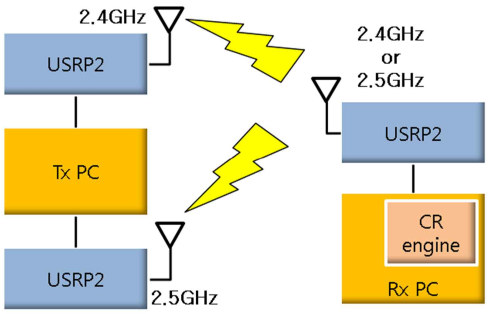


Figure 7. Test scenario for CR engine

Figure 7 illustrates a test scenario for verification of the CR engine implemented using UHD library. Video stream data to be transmitted from the transmitting PC are encoded by the LTE UL software modem implemented in GPU. Then, the data are modulated with USRP2 of which the center frequency is 2.4GHz and 2.5GHz, respectively. In the receiving PC we first set up the receive frequency to be 2.4GHz. When the signal transmission with 2.4GHz is suspended and the center frequency is changed to 2.5GHz, we checked if the center frequency of the receiving USRP2 is also changed from 2.4GHz to 2.5GHz in accordance with the frequency change at the transmitter.

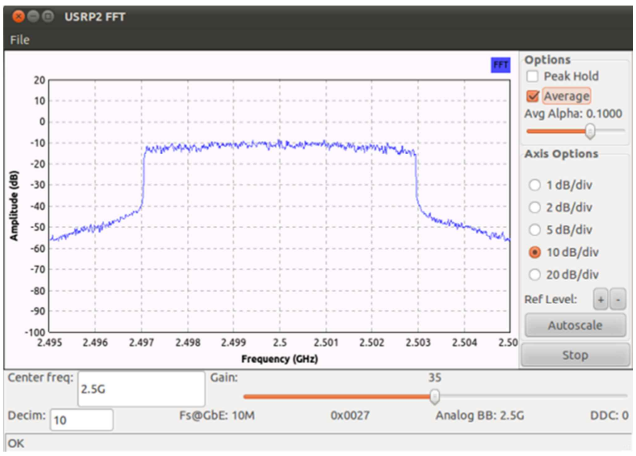


Figure 8. Spectrum of received signal in USRP2

Figure 8 illustrates the spectral domain power distribution of received signals after the receive frequency has been changed from 2.4GHz to 2.5GHz according to the CR engine introduced in Section 3.3. The spectrum measurement has been obtained using *usrp2_fft.py* program provided by GNU Radio. As shown in Figure 7, the center frequency is changed from 2.4GHz to 2.5GHz in accordance with the change in the transmitter.

5. CONCLUSION

In this paper, we implemented an SDR-based LTE UL system using GPU and USRP2 as its modem part and RF transceiver part, respectively. Baseband signal processing part has been modified properly for exploiting as many threads as possible using parallel processing on GPU. In addition, using the library provided by UHD, CR engine has been implemented for switching the center frequency automatically in accordance with the change of RF frequency in the transmitter.

In order to verify the implemented system, we measured the processing time required by baseband signal processing, which turned out to be 5.00ms and 6.79ms in the case of transmitter and receiver, respectively. We also verified that the video stream data can be retrieved at the receiver through the experimental tests of real-time transmission and reception. Finally, we checked that our proposed CR engine traces the change of RF frequency from 2.4GHz to 2.5GHz. It particularly means that the proposed CR engine can automatically search for a best appropriate waveform when the current one is somehow to be switched with another waveform. We are now trying to expand our CR engine, which exploits the various applications with UHD libraries, to the MIMO system.

ACKNOWLEDGEMENT

This research was supported by the MKE (The Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program (NIPA-2012-H0301-12-4003) supervised by the NIPA (National IT Industry Promotion Agency).

REFERENCES

[1] June Kim, Seungheon Hyeon, and Seungwon Choi, “Implementation of an SDR System Using Graphics Processing Unit”, IEEE Communication Magazine, Vol. 48, pp. 156 -162, 2010.

[2] Peter B. Jorgensen, Thomas L. Hansen, Troels B. Sorensen, and Gilberto Berardinelli, “Implementation of LTE SC-FDMA on the USRP2 Software Defined Radio Platform”,

IEEE Swedish Communication Technologies Workshop, 2011.
[3] NVIDIA Corporation, CUDA Programming Guide, NVIDIA Corporation, Apr. 2009.
[4] Ettus Research, Change log for releases. http://ettus-apps.sourcerepo.com/redmine/ettus/projects/uhd/wiki/ChangeLog.
[5] 3G Generation Partnership Project (3GPP); Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Channels and Modulation (Release 9), http://3gpp.org/ftp/specs/html-info/36211.htm
[6] 3G Generation Partnership Project (3GPP); Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA); Multiplexing and channel coding (Release 9), http://3gpp.org/ftp/specs/html-info/36212.htm
[7] NVIDIA Corporation, NVIDIA GTX 295 Datasheet. http://www.nvidia.com/object/product_geforce_gtx_295_us.html.
[8] Chiyoung Ahn, Saehee Bang, Hyohan Kim, Seunghak Lee,June Kim, Seungwon Choi, and John Glossner, "Implementation of an SDR system using an MPI-based GPU cluster for WiMAX and LTE", Analog Integrated Circuit and Signal Processing, Springer, 2012.