PERFORMANCE EVALUATION OF FRONT-END SIGNAL PROCESSING RECEIVER FOR ISDB-T 1SEG SOFTWARE DEFINED RADIO

Tomonori MIYANO and Tomohisa WADA

Graduate School of Engineering and Science,
University of the Ryukyus, 1 Senbaru Nishihara Okinawa, 903-0213, Japan

ABSTRACT

ISDB-T (Integrated Services Digital Broadcasting - Terrestrial), that is television format in Japan, has comparatively-easily processed 1seg system. Digital Signal Processing (DSP) of 1seg receiver is normally implemented by Hardware Defined Radio (HDR). In this paper, front-end part of 1seg receiver system is implemented by Software Defined Radio (SDR), which is used by GNU Radio, and evaluated CPU utilization.

Index Terms— ISDB-T, 1seg, Software Defined Radio, GNU Radio, OFDM, Low Pass Filter, Resampler

1. INTRODUCTION

ISDB-T specification [1] is utilized for the digital terrestrial television broadcasting service in Japan. The feature of ISDB-T is to simultaneously support both High Definition Television (HDTV) for fixed application and 1seg service for low quality but mobile application such as cellular phone television receivers. Digital wireless signal processing of 1seg is conventionally implemented by hardware digital circuit. Since PC computation power is increasing year by year, software based wireless communication system will be strong application for PC platform. In this paper, SDR [2] prototype for Orthogonal Frequency Division Multiplexing (OFDM) which is used in 1seg receiver is investigated because of its comparatively-low computation requirement. In order to clarify the analog and digital interface problem, GNU Radio[3] frame and Universal Software Radio Peripheral 2 (USRP2) [4] interface box, which is supporting GNU Radio, is used and evaluate CPU performance for 1seg front-end signal processing.

The rest of paper is organized as follows. Section 2 provide GNU Radio and ISDB-T architecture. Section 3 describe specification and evaluation result of 1seg OFDM front-end system. Then conclusion is given in Section4

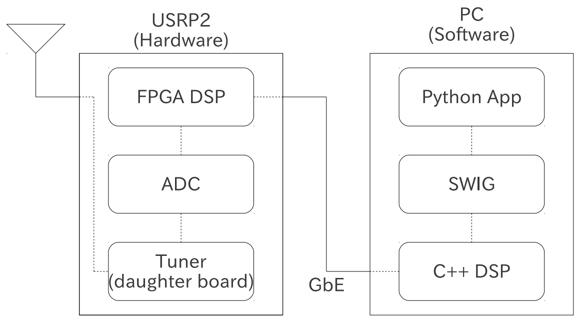

Fig. 1. GNU Radio Architecture (Receiver)

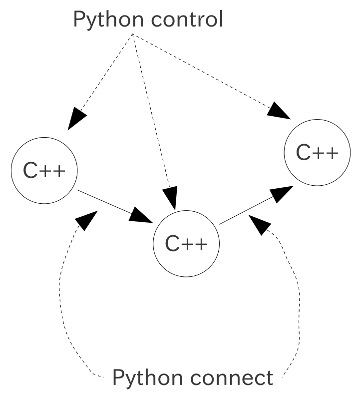

Fig. 2. GNU Radio Programming

2. SDR ARCHITECTURE

2.1. GNU Radio

GNU Radio is an open source software collection aimed at developing SDR. Fig.1 shows a simple architecture model for GNU Radio. This architecture consists of two parts: software and hardware.

Software part is implemented by Python and C++ programming languages on PC. This software can be available on Linux, Mac OS X, and Windows. C++ and Python programming languages are connected by using Simplified Wrapper and Interface Generator (SWIG) [5] which is a software de-

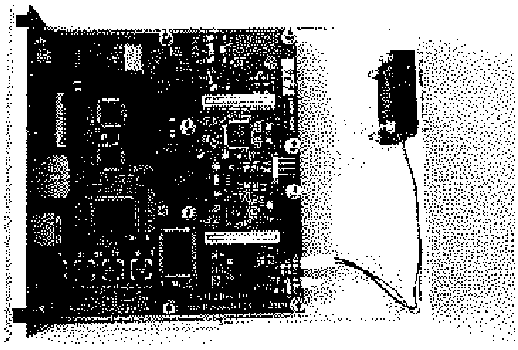

Fig. 3. USRP2(Motherboard)

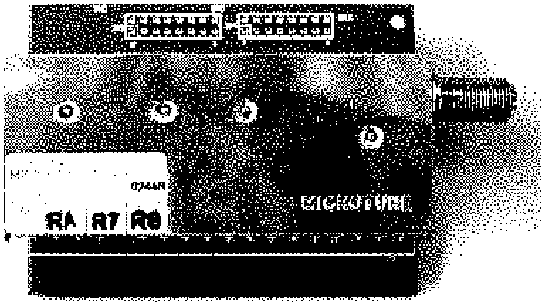

Fig. 4. TVRX(Daughterboard)

Table 1. USRP2 Specification

USRP2(Motherboard)	
FPGA	Xilinx Spartan 3-2000
ADC	Dual 100Msps 14-bit
DAC	Dual 400Msps 16-bit
SRAM	1MByte
Interface	2Gbps Serial Interface
TVRX(Daughterboard)	
Minimum Frequency	50MHz
Maximum Frequency	860MHz
Bandwidth	6MHz
Noise Figure	8dB
Minimum Gain	0
Maximum Gain	95

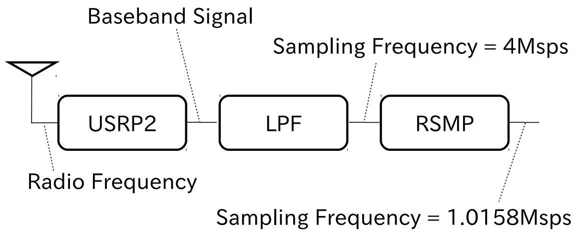

Fig. 5. Front-end Signal Processing

velopment tool. As shown in Fig. 2, C++ programming language is used for software defined DSP. On the other hand, Python programming language connects each C++ module, and control software and hardware defined DSP.

Hardware part is implemented by USRP2. USRP2 is the device supported by GNU Radio, and has two sets of the board such as motherboard as shown in Fig.3 and daughterboard called TVRX as shown in Fig.4. USRP2 and PC are connected by Gigabit Ethernet (GbE). Motherboard works Analog to Digital Converter (ADC), Digital to Analog Converter (DAC), and a large Field Programmable Gate Array (FPGA) which is optimized for DSP.Higher sample rate processing like Digital Down Conversion (DDC) and Digital Up Conversion (DUC) is handled in FPGA, and lower sample rate processing can be performed on PC. Daughter board carries analog tuner component for Radio Frequency (RF) part. There are several different types of daughterboard with using frequency band. TVRX is a VHF and UHF receiver system on a TV tuner. Table 1 shows USRP2 and TVRX specification.

2.2. Front-end Signal Processing Receiver

Fig.5 shows block diagram of front-end signal processing of the 1seg receiver. ADC and DDC converts RF signal to baseband signal on USRP2. Maximum 100Msps of sampling frequency is supported by USRP2. However in our prototype much less sampling frequency is used since General Purpose Processor (GPP) cannot process such high sampling data. Then, decimation function in USRP2 is used to reduce sampling frequency. Decimation rate can be programmable by Python.

Since TVRX daughter board supports only 6MHz channel width and 1seg signal bandwidth is roughly 430KHz, baseband signal from USRP2 bandwidth has to be reduced by Software Low Pass Filter (LPF). LPF output corresponds to required 1seg OFDM signal. There are Window Function and Remez Exchange Method [6] as a function of filter of GNU Radio. It is easy to design filter by using Window Function, but cannot be optimum approximation. GNU Radio have follow five Window Functions.

- Rectangular Window
- Hann Window
- Hamming Window
- Blackman Window
- Kaiser Window

On the other hand, Remez Exchange Method can optimize transfer function which is machined to specifications and is the lowest taps. It is used in this paper. Fig.6 shows the spectrum output from USRP2 before LPF. This specifications

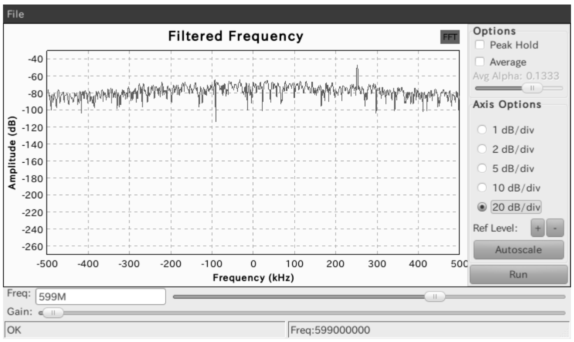

Fig. 6. Spectrum after USRP2

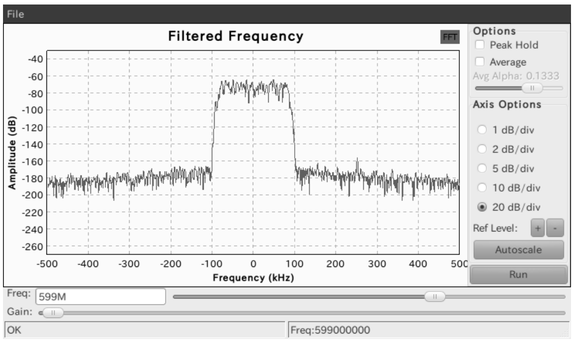

Fig. 7. Spectrum after LPF

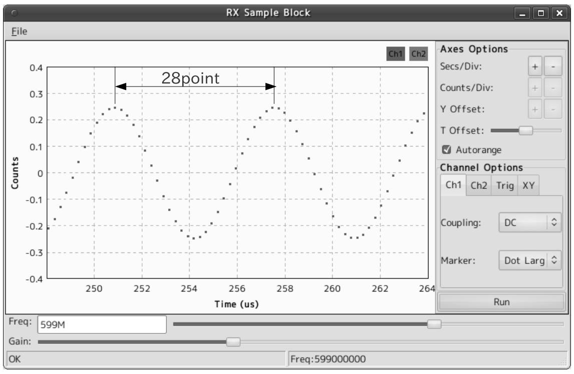

Fig. 8. Time Signal before RSMP

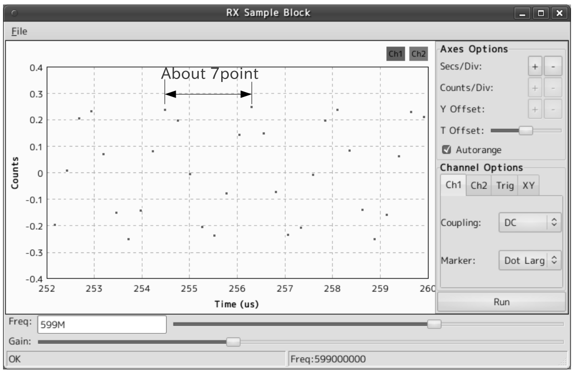

Fig. 9. Time Signal after RSMP

shows in Table.2. Like AWGN model, entire frequency are similar intensity. Fig.7 shows that frequency band required for 1seg receiver is extracted by LPF.

Table 2. LPF Specification

Decimation Rate	25
Frequency Band	599MHz
Gain	1

Resampler (RSMP) change sampling frequency [7]. Main demodulation process of 1seg OFDM is 1K sample FFT (Fast Fourier Transform) and the valid symbol length is $1008\mu sec$. This sampling frequency is $1024/1008\mu = 1.0158MHz$. Thus it is necessary to convert that from sampling frequency FsMHz before resample to 1.0158MHz. Then, decimation rate is FsMHz, interpolation rate is 1.0158MHz and sampling rate is 1.0158/Fs. This sampling rate decide filter design. Fig.8 shows sine wave with 4MHz. 28points per cycle can be confirmed. While Fig.9 shows sine wave converted from 4MHz to 1.0158MHz by RSMP. It is about 7points per cycle and is converted to be a quarter of points before RSMP.

3. EVALUATION

The front-end system shown in Fig.5 is created by Python program, and is evaluated amount of CPU used for 1 minute.

3.1. SPECIFICATION

Fig.10 shows evaluation environment. Left side USRP2 is connected to the right PC thorough GbE. The PC specification is shown in Table 3. USRP2 has decimation rate is integer value which divides 100Msps. Decimation rate is adopted 50 because of the closest value 100Msps/50=2Msps to 1.0158Msps after RSMP. Frequency is 599MHz which used as one of TV channel.

Table 3. Simulation Specification

OS	Ubuntu 9.10
CPU	Intel Core 2 Duo 3.06GHz
Memory	4.00GB
Frequency	599MHz
Decimation Rate	50

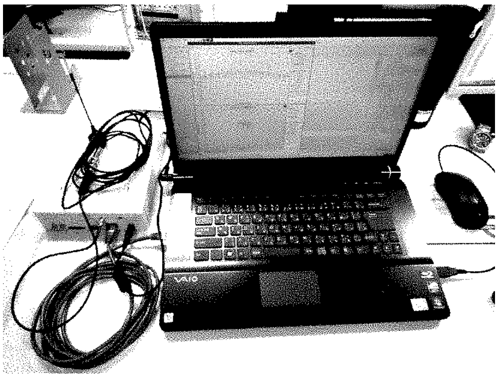

Fig. 10. My Setup

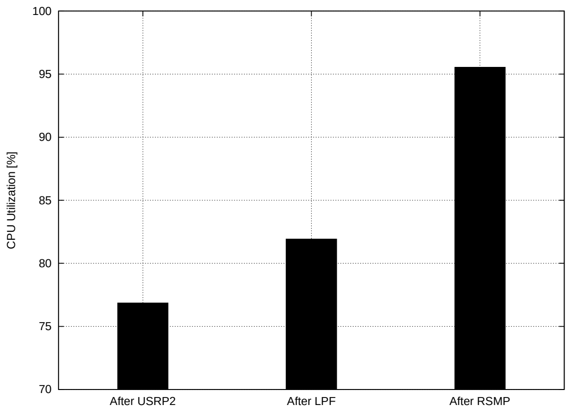

Fig. 11. CPU Utilization

3.2. RESULT

Fig.11 shows the measurement of the average CPU utilization on 1seg front-end system. Maximum of CPU utilization is 200% because of using dual core. Left bar is CPU utilization of sending from USRP2 to PC. Middle is extended LPF on left, and right is additionally extended RSMP on middle. CPU utilization increased 5% between left and middle. In addition, right is increased more than 10% middle. This is considered because the filter of RSMP is complex process.

4. CONCLUSION

In this paper, 1seg front-end process is implemented using GNU Radio. Radio wave is converted to digital signal through USRP2, channel for 1seg broadcasting is taken and sampling frequency is changed by RSMP. And then CPU utilization is evaluated. Total CPU utilization of this system is 95.5%. In the future research, RSMP will be improved and other 1seg process will be implemented.

5. REFERENCES

[1] *Transmission system for digital terrestrial television broadcasting*, Asociation of Radio Industries and Business, September 2007.

[2] Paul BURNS, *Software Defined Radio for 3G*, Artech House Publishers, 2002.

[3] “GNU Radio,” http://gnuradio.org.

[4] “Ettus resarch llc,” http://www.ettus.com.

[5] “SWIG:Simplified Wrapper and Interface Generator,” http://swig.org.

[6] T.W.Parks and J.H. McClellan, “Chebyshev approximation for nonrecursive digital filters with linear phase,” *IEEE Trans. Circuit Theory*, vol. CT-19, no. 2, pp. 189–194, 1972.

[7] Daigo KASHIMA, Tomohisa WADA, and Shuji MURAKAMI, “Investigation on software radio design for front-end signal processing of terrestrial ofdm receiver,” *Technical report of IEICE. ICD*, vol. 103, no. 705, pp. 25–30, 2004-03-05.