Using SDR for Cost-Effective DTV Applications

J. Kwak, Y. Park, and H. Kim
Dept. of Computer Science and Engineering, Korea University, Seoul, Korea
{jwuser01, ytpark, hyogon}@korea.ac.kr

Abstract – ***Today's digital TVs can provide advanced functionalities such as DVR, EPG, and PIP. They require simultaneous access to multiple channels using additional hardware resources, which increases the cost. In order to solve this problem, we propose to use wideband software defined radio (SDR) that can demodulate multiple channels on a single radio frontend. Through the SDR, we can obtain multiple channels from a single wideband using frequency translation and FIR filter. Specifically, we use only one radio frontend and obtain multiple MPEG-TSs. Through this approach, digital TV applications can be dynamically and flexibly added to or deleted from smart TVs without incurring hardware cost. We demonstrate a prototype built on USRP and GNU Radio that works on real broadcasting streams.***

Keywords: Broadcasting, Software defined radio (SDR), Multiple channels, Digital television applications

1 Introduction

Viewers' demand for higher resolution and better visual quality has led us to Ultra High Definition (UHD) broadcasting. It inevitably increases the volume of data to deliver over air, so video coding and transmission technologies have significantly evolved. Still, flexible band allocation and modulation are required for efficient use of wireless resource use. But existing Digital Television sets (DTVs) have used hardware-based demodulator, making dynamic band allocation and modulation scheme changes difficult. This problem sparked interests in software programmable radio (SDR) to replace the hardware demodulator. The challenges in the attempt, however, are the high cost of SDR devices and the computing power required to perform signal processing on broadcast streams.

What is encouraging is that recent smart TVs are equipped with high-performance processors such as quad-core CPUs. They enable SDRs to process broadcast streams in real time. For instance, GNU Radio 3.7.5 on Intel i7 processor could receive, demodulate, and display a HDTV stream in real time [1]. Therefore, a smart TV set can lend itself to SDR for novel DTV applications, even though they do not have a dedicated SDR device. Some DTV applications available today are as follows:

- Digital Video Recorder (DVR): stores digital format video on a USB flash drive, SD memory card, hard disk, or networked mass storage device.
- Electronic Program Guide (EPG): continuously updates and displays broadcast program and schedule information on the TV screen. The program information is conveyed through each channel. Some DTVs can gather the program information on background while the user is watching a program.
- Picture in Picture (PIP): displays a small inset window showing other channel while playing a channel on the main screen.

These applications operate by obtaining data from multiple channels in parallel. Moreover, each application can tune to different channels. In order to process a channel, we need a chain of processing blocks from RF antenna to data randomizer, which is depicted in Figure 1[2]. Typically, a hardware logic implements these blocks.

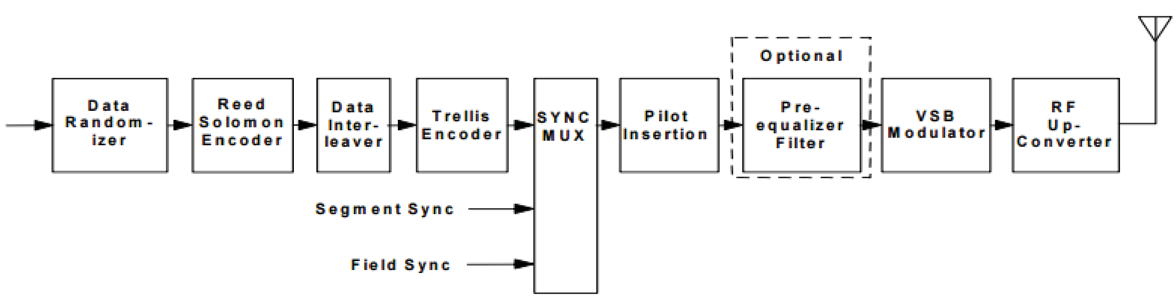

Figure 1 DTV functional block diagram

Because these applications add to the value of the TV sets, manufacturers would want them on all models. But the prohibitive cost of necessary hardware only allows them to provide the applications on high-end models. Specifically, two or more channel demodulation hardware modules are added to the TV set. But in this paper, we aim to expand the range of models that support the advanced DTV applications by using SDRs. In particular, we demonstrate that existing smart TVs can exploit the SDR technology so that it can demodulate multiple channels without using as many hardware demodulators. Through this approach, we hope that we can add values to smart TVs without incurring excessive cost.

2 Related Work

2.1 SDR in Broadcasting Technology

There has been continual research on SDR technology to apply it to the broadcasting system. It involves difficulties because SDR requires high computing power for signal processing. Consequently, it could not be applied to high-bit rate broadcast systems. For instance, the second generation terrestrial digital video broadcasting (DVB-T2) needs the bit rates between 7.44 Mbit/s and 50.32 Mbit/s, and the Advanced Television Systems Committee (ATSC) 8-level Vestigial Sideband Modulation (8-VBS) uses 19.39Mbit/s [2].

So far, researchers have managed to implement DVB-T/H on MuSIC (Multiple SIMD Cores) platform [3], and DVB-T on GNU Radio [4]. They are testing with various SDR platforms as well, to meet the required performance for real-time applications. For instance, there is a DVB-T2 implementation on CGRA processor [5,6]. There is also a DVB-C2 implementation [7]. But these efforts focus on implementing basic broadcasting specifications and performance enhancements. In this paper, we aim to extend the SDR technology to perform multiple channel demodulation for the DTV applications that require simultaneous processing of multiple channels.

2.2 Multiple channel demodulation using wideband SDR

A SDR device can obtain multiple narrowbands from a wideband using a single radio frontend if the radio front and the processing capability allows it. The radio front can use a single analog filter and analog-to-digital converter (ADC), and no additional hardware cost is incurred in filtering multiple channels [8,9]. Albeit in different domains, such techniques have been explored. For instance, packet capturing can be simultaneously done on multiple IEEE 802.15.4 channels [10,11]. The technique is also utilized in finding the best frequency in chaotic cognitive radio environment [12]. But it has not been applied to broadcasting systems.

3 System Architecture

Our system is built on the Universal Software Radio Peripheral (USRP) platform using GNU Radio. GNU Radio is an open source SDR development system for terminals [13]. It provides a collection of various signal processing blocks, such as ATSC and DVB that are essential to the DTV pipeline. Since GNU Radio can connect the signal processing blocks using either C++, Python, or a graphic tool, it facilitates flexible development. Figure 2 shows the SDR implementation of the block diagram in Figure 1 that uses the ATSC pipeline in GNU Radio. This pipeline in Figure 2 produces a MPEG Transport stream (MPEG-TS). MPEG-TS is the standard container format to store audio, video, and Program and System Information Protocol data [14]. It is widely used in such broadcasting systems as DVB, ATSC, Integrated Services Digital Broadcasting (ISDB), and Internet Protocol television (IPTV).

USRP is a SDR hardware designed for GNU Radio Project by Ettus Research [15]. It passes the I/Q samples that passed the RF frontend and ADC to host PC through USB or other communication technologies in a stream. The host PC processes the stream using GNU Radio.

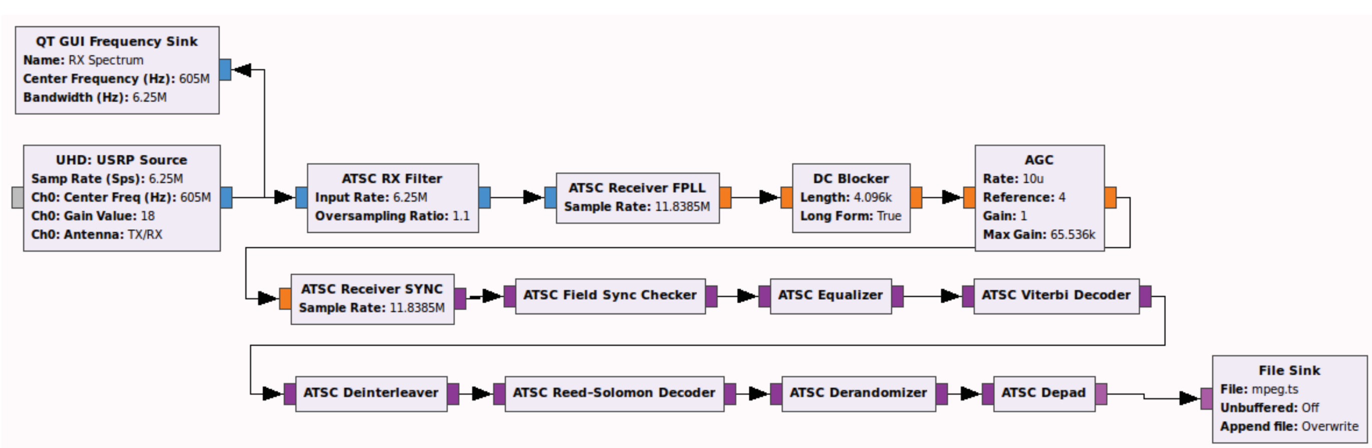

Figure 2 ATSC DTV pipeline of GNU radio

ISBN: 1-60132-440-5, CSREA Press ©

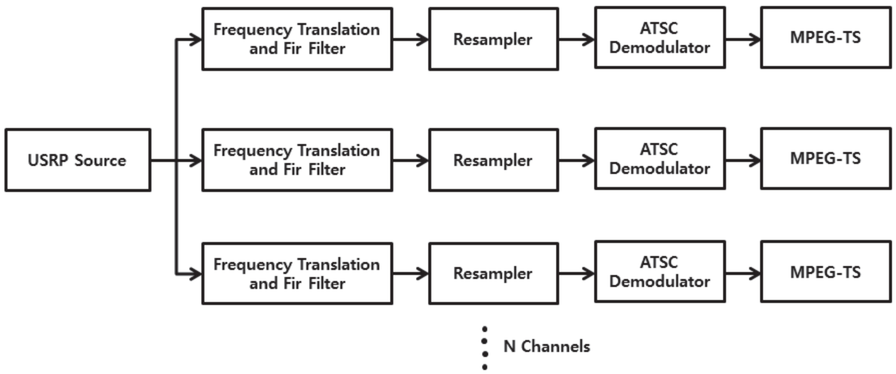

Figure 3 System architecture

ATSC-Terrestrial is transported using VHF low-band (54~88 MHz), VHF high-band (174~216 MHz), and UHF band (470~890 MHz) where each channel uses 6 MHz. To demodulate *N* channels, we receive the wideband at the analog front end and then pass each channel through frequency translation, FIR filter, Resampler, and ATSC demodulator. It produces a MPEG-TS for each channel.

Frequency Range	70 MHz – 6 GHz
ADC Resolution	12 Bits
Max Bandwidth	56 MHz
TX/RX	2 x TX and 2 x RX (Full Duplex)
Preselectors	None

Table 1 USRP B210 features

We used GNU Radio 3.7.9 release and USRP B210 to run our implemented system. The specifications for USRP B210 are given in Table1. The device can receive a UHF band, and can simultaneously receive up to nine channels in case of ATSC-Terrestrial through wideband reception.

As we mentioned in the Introduction section, relatively high-end PC processing power can afford demodulate a single channel in real time. Therefore, in our experiment we store the I/Q sample data from the USRP into a file and produced the output offline.

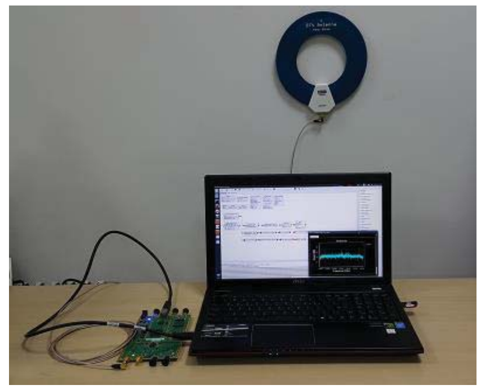

Figure 4 Test setup

We implemented the system shown in Figure 4 using GNU Radio. It receives a 18 MHz band and demodulate two channels therein. After translating the target frequency range through Frequency Xlating FIR filter, it extracts the channels through Low Pass Filter. Then it adjusts the sample rate through Fractional Resampler, and demodulate through ATSC receive pipeline. Figure 5 shows our test setup. It is composed of a RF antenna that can receive ATSC Terrestrial, a host PC, and USRP B210.

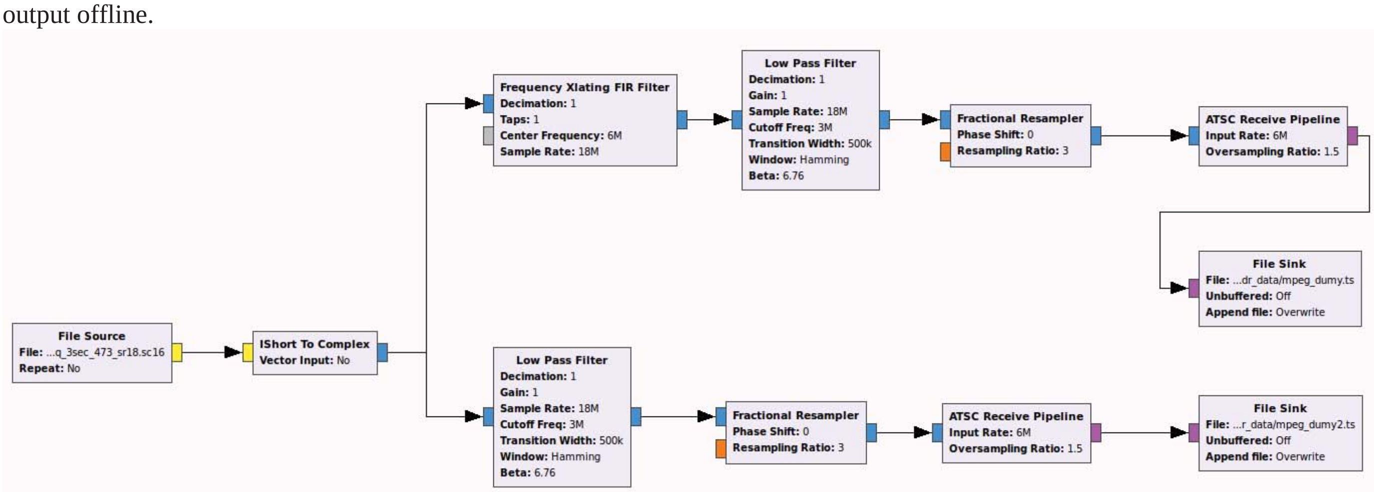

Figure 5 Implement of multi-channel broadcasting receiver

4 Evaluation

We used real broadcasting signals for our test. In Seoul, Korea, each channel used in the experiment is allocated 6 MHz over 470 MHz (channel 14) ~ 500 MHz (channel 18).

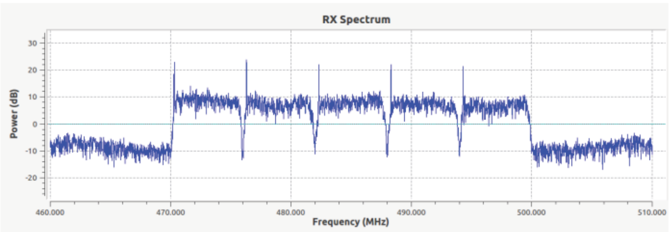

Figure 6 Broadcasting frequency band in Seoul

Figure 7 shows the process of extracting the signal of Channel 15 in our system. It confirms that our system effectively obtains the target signal by applying Frequency Xlating FIR Filter, Low pass filter, and Fractional Resampler Block to the wideband signal. We can also check that the processing has been done correctly by playing the extracted channels on Media Player (Figure 8 and 9) and by using PISP parsing tool (Figure 10).

Figure 7 Video of channel 14

Figure 8 Video of channel 15

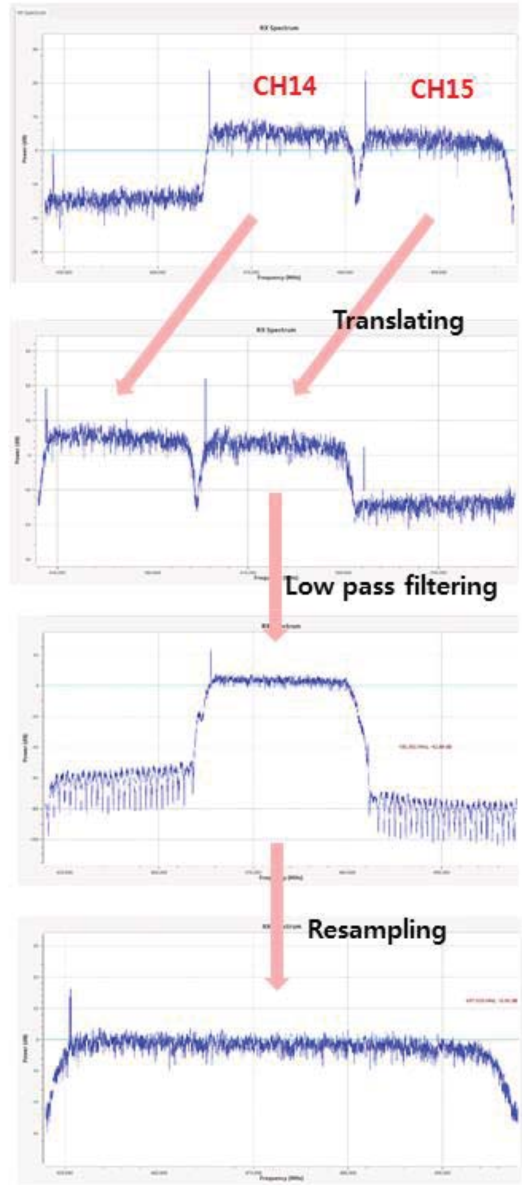

Figure 9 Filtering of channel 15

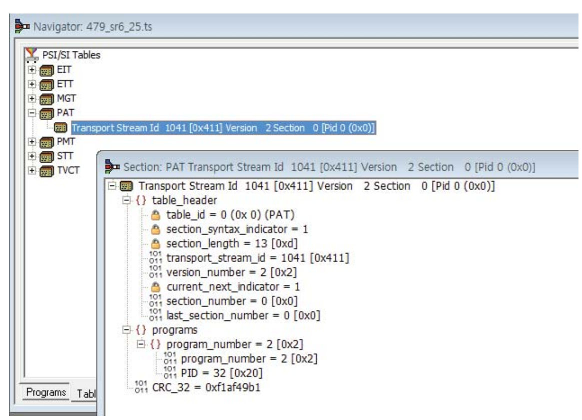

Figure 10 PSIP data of channel 15

The implemented system can demodulate multiple channels through software pipeline without using extra hardware. But it incurs computing cost to extract multiple channels. Figure 11 shows the CPU utilization increases in the number of filtered channels when the wideband has a width of 30 MHz. This is because the GNU Radio filter implementation depends on computation in the CPU. In order to resolve this heavy computation issue, researchers are eyeing graphics processing units (GPUs) to offload the computing intensive FIR Filter [16,17].

Figure 12 measures the CPU utilization when we increase the width of the wideband from 1 to 5 times the channel width. The result shows that the utilization increase is small in the width of the wideband. It tells us that as long as the transmission speed between the SDR device and the host PC is sufficient, we can process a larger wideband without proportionally increasing the CPU utilization. This aspect makes the SDR approach to DTV applications very attractive.

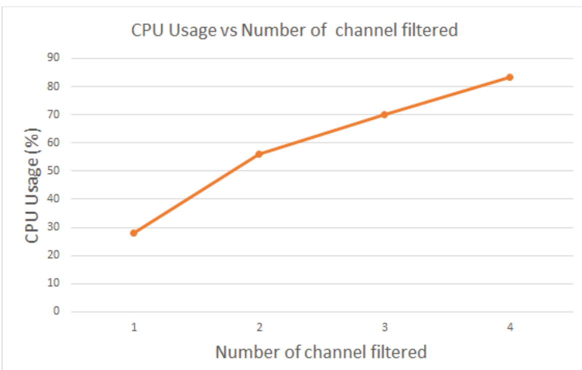

Figure 11 CPU Usage vs Number of channel filtered

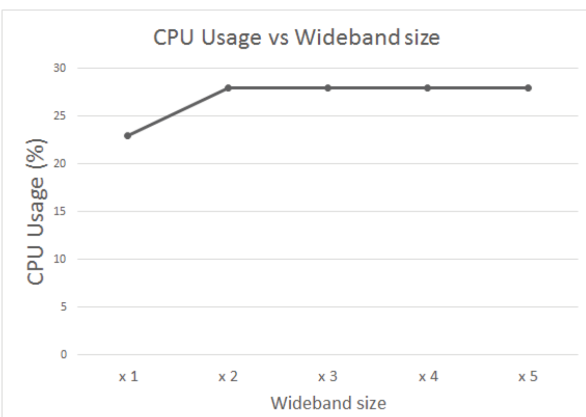

Figure 12 CPU Usage vs Wideband size

5 Conclusions

This paper shows that we can receive and demodulate multiple DTV channels using GNU Radio and USRP without adding extra hardware. It allows applications such as DVR, PIP, and EPG with SDR. Moreover, it can enable auto channel scan and channel switch latency reduction because we can minimize the switch in the analog frontend. Although SDR is still experimental for broadcasting due to cost and performance issues, the decrease in computing cost will offset the cost and make SDR a viable solution for future smart TVs.

6 References

[1] GNU Radio release-3.7.5 note [Online] - https://gnuradio.org/redmine/versions/43.

[2] Advanced Television Systems Committee, "ATSC Digital Television Standard Part 2 RF Transmission System Characteristics (A/53, Part 2: 2007)".

[3] Y. Jiang, W. Xu, and C. Grassmann, "Implementing a DVB-T/H receiver on a software-defined radio platform", International Journal of Digital Multimedia Broadcasting vol. 2009.

[4] V. Pellegrini, G. Bacci, and M. Luise, "Soft-DVB: a fully-software GNURadio-based ETSI DVB-T modulator", 5th Karlsruhe Workshop on Software Radios, 2008.

[5] H. Yang, *et al*, "Software-defined DVT-T2 demodulator using scalable DSP processors", Consumer Electronics, IEEE Transactions, vol. 59, no. 2, pp. 428-434, 2013.

[6] C. Kocks, *et al*, "A DVB-T2 receiver realization based on a software-defined radio concept", Communications, Control and Signal Processing, 4th IEEE International Symposium, 2010.

[7] P. Hasse and J. Robert, "A software-based real-time DVB-C2 receiver", Broadband Multimedia Systems and Broadcasting, IEEE International Symposium, 2011.

[8] L. Pucker, "Channelization techniques for software defined radio", Proceedings of SDR Forum Conference, 2003.

[9] H. Tsurumi and Y. Suzuki, "Broadband RF stage architecture for software-defined radio in handheld terminal applications", Communications Magazine, IEEE vol. 37, no.2, pp. 90-95, 1999.

[10] L. Choong, "Multi-channel IEEE 802.15. 4 packet capture using software defined radio", Networked & Embedded Systems Laboratory, UCLA, Technical Report TR-UCLA-NESL-200904-01, 2009.

[11] R. Matos, A. Fröhlich, and L. Becker, "Using multiple channels to improve SDR flexibility and performance", Computing, Networking and Communications, IEEE International Conference, 2012.

[12] R. Zhou, *et al*, "Software defined radio based frequency domain chaotic cognitive radio", SOC Conference, IEEE International, 2011.

[13] GNU Radio [Online] - http://www.gnuradio.org/.

[14] Recommendation, I. T. U. T. H. "222.0 (2006)| ISO/IEC 13818-1: 2007", Information technology–Generic coding of moving pictures and associated audio information: Systems, 2007.

[15] Ettus Research [Online] - http://www.ettus.com/.

[16] K. Moreland and E. Angel, "The FFT on a GPU", Proceedings of SIGGRAPH/Eurographics Work-shop on Graphics Hardware, 2003.

[17] A. Smirnov and T. Chiueh, "An Implementation of a FIR Filter on a GPU", Experimental Computer Systems Lab, Stony Brook University, Tech. Rep, 2005.