A Real-Time PC Based Software Radio DVB-T Receiver

Shu-Ming Tseng
Department of Electronic Engineering
National Taipei University of Technology
Taipei, Taiwan
shuming@ntut.edu.tw

Jian-Cheng Yu
Department of Electronic Engineering
National Taipei University of Technology
Taipei, Taiwan
g931303@hotmail.com

Yueh-Teng Hsu
LiteOn Technology Cooperation,
Taipei, Taiwan
Matthew.Hsu@liteon.com

***Abstract*—The total data rate of DVB (Digital Video Broadcasting) is 9.953Mbps much greater than DAB (Digital Audio Broadcasting). It is necessary to use better signal processing algorithm to improve the performance of DVB receiver. Hence, we propose some methods: (1) To optimize Reed-Solomon and Viterbi decoder; (2) To use parallel process instructions; (3) To Reduce FOR Loop and IF Branch, etc. After these modifications, the software radio in real-time reception based on PC can be implemented. In summary, we demonstrate a software receiver which operates at an acceptable real-time decoding speed.**

Keywords-software radio; mobility; Digital Video Broadcasting; orthogonal frequency-division multiplexing.

I. INTRODUCTION

The SR (Software Radio) based on PC platform has become popular and important in software GPS (Global Position system) [1] and software DAB (Digital Audio Broadcasting) [2]. Besides, the parallel process is one of the important and useful methods to enhance the decoding speed in SR. It is possible to use the software to implement the demodulation and decoding at real time based on PC. The total data rate of DAB is 2.4 Mbps (DQPSK); DVB-T (Digital Video Broadcasting-Terrestrial) is 9.953 Mbps (16QAM, guard interval 1/4, rate 2/3 convolutional code, non-hierarchical system for 6 MHz channels). Thus, the data rate of DVB-T is much greater than DAB. We hope to implement the SR which is used for DVB-T signal in Taiwan [3]. The advantages and benefits of SR are multimode and low cost. Besides, it is easy to maintain and modify. As we know, it is almost impossible to manufacture a new ASIC (Application-Specific Integrated Circuit) chip, which is relatively very expensive and time-consuming. On the contrary, the SR receiver does not need any additional hardware. The most important benefit of SR research, when compared with hardware radio, is that people can modify the signal processing procedure, algorithm and test the result easily. People can use the better detecting or estimation algorithm with software language to improve the performance of radio receiver rather than producing a new ASIC chip. For real-time DVB-T reception, we need to improve the slowest part first. Therefore, we have to implement the software of the RS (Reed-Solomon) and Viterbi decoder [4], which is fast enough to decode in real time. In order to enhance the SR decoding performance, we use the following ways: Looking up tables to reduce operation of program and SIMD (Single Instruction Multiple Data) instructions to decrease the CPU operation time [5], etc. Meanwhile, the performance of Viterbi decoder also needs to be improved by several methods: (1) To use SIMD instructions; (2) To reduce the branch of the program; (3) The other efficient methods [4].

This paper is organized as follows: We describe the previous SR in Section I. The proposed system architecture is described in details in Section II. Implementation and results are described in Section III. Section IV presents the conclusions.

II. SYSTEM ARCHITECTURE

The basic components for a software research platform include a PC and a radio front-end. The architecture is shown in Figure 1. It has hardware and software parts. The software structure is comprised of baseband processing, MPEG2 audio decoder and test application. The system memory acquires the digitized IF (Intermediate Frequency) signal from the tuner device by the driver; this is so called raw data. When enough raw data is collected, we can calculate the SNR and do baseband processing or dump that into the hard disk for future analysis. Thus, we can get the transport stream after baseband processing.

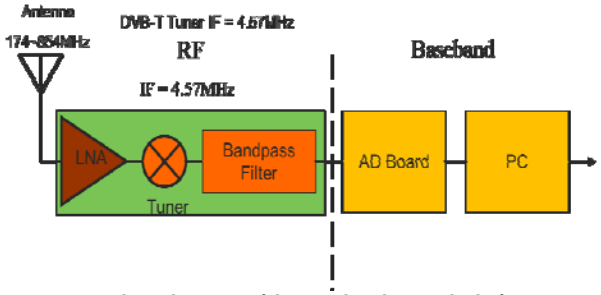

Figure 1. The architecture of the introduced research platform.

 ISBN: 978-1-61208-154-0

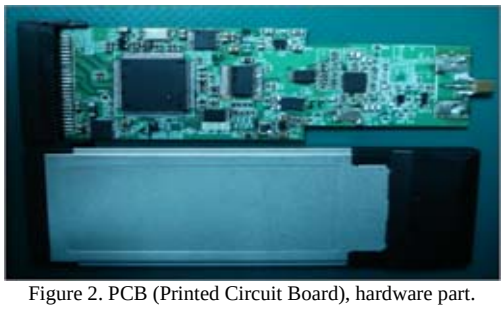

Figure 2. PCB (Printed Circuit Board), hardware part.

A. *Hardware*

The hardware part is shown in Figure 2. The receiver can decode the radio signal and record the baseband signal into hard disk simultaneously. It has several parts in our AD board. This device is comprised of A/D chip and LVDS (low voltage differential signaling) to reduce the transmission interference. The architecture is shown in Figure 3. The tuner module has relations with the receive sensitivity. Low IF signal output amplify gain is suitable for the requirement of analog to digital converter and so on. In our design, the silicon tuner (Xceive xc3028) is chosen to fit the requirements. The TLI 5540 is a high-frequency signal processing IC and converts the received frequency to IF frequency (4.571MHz). The frequency command is fed through the I^2C (I-squared-C) interface on this tuner. According to the DVB-T spec: the sampling rate of 8 MHz is 64 MHz /7= 9.142858 (MHz). Thus, we use the (64/7) x3 = 27.42857 (MHz) for 8bits resolution sampling rates to be digitalized by an AD converter and is fed into PCI-Express (bridge) chip (Transfer rate > 1 gigabyte per second) and can be sent to PC for future processing.

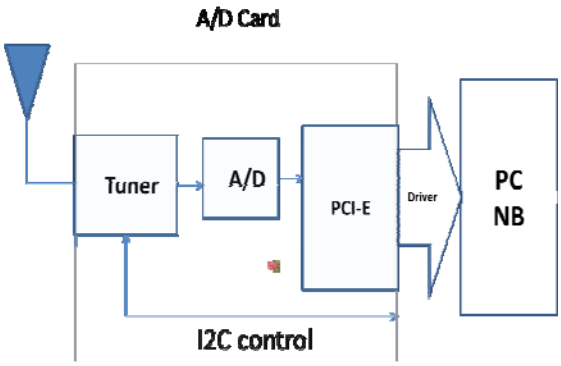

Figure 3. The architecture of the hardware part.

TABLE I. TUNER CONFIGURATION.

Parameter	Value
Frequency Input range	174~854 MHz
Bandwidth	6/7/8 MHz
Method of down conversion	Up down convertor

B. *Software*

The software has two parts. One is baseband receiver, and the other is baseband data recorder. The signal processing of software structure is shown in Figure 4 and 5, which is described as below: The purpose of time synchronization is used to synchronize the symbol in the FFT (Fast Fourier Transform) window correctly. We use the CP (Continual Pilot) to estimate integral frequency offset and use the TPS (Transmission Parameter Signaling) pilot to estimate the offset of OFDM (Orthogonal Frequency-Division Multiplexing) symbol. The Fractional frequency Synchronization can modify the signal after digital down converter in correct baseband frequency. We will use scatter pilot to estimate the channel coefficients. After doing fractional frequency synchronization and integer frequency synchronization, the residual frequency offset still exists. However, we can utilize a mathematic model for an offset with sampling frequency offset and residual frequency. We have to do channel estimation to compensate the phase error. The DVB-T system uses TPS to do frame synchronization.

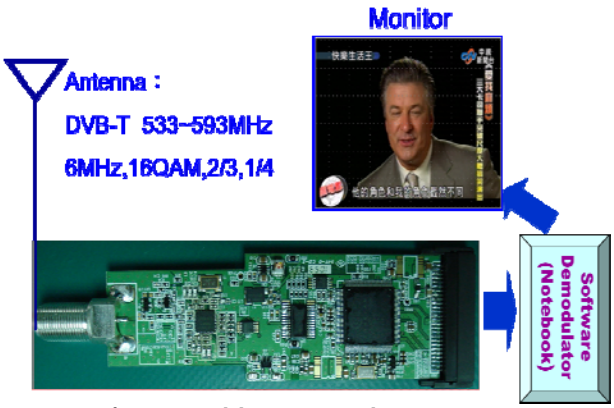

Figure 4. Software Demodulator operation diagram

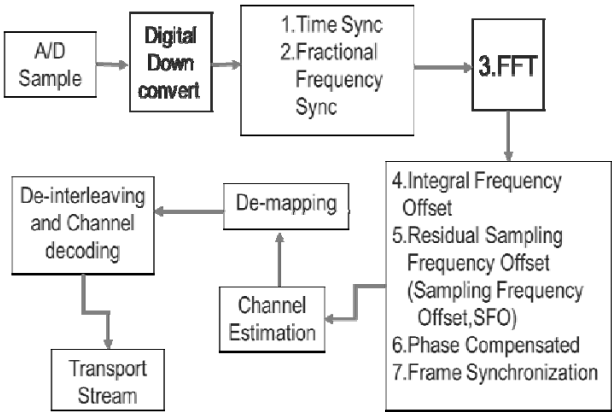

Figure 5. Structure of BDA(Broadcast Driver Architecture) filters

1) Sampling Frequency Offset: It will generate SFO (Sampling Frequency Offset) when the baseband sampling

rate of receiver cannot match the transmitter. Thus, we have to use some algorithms to modify SFO [6]-[7].

$$\delta = \frac{\delta_a}{T_s}$$
$$T_s' = T_s(1+\delta) \quad (1)$$

The u_{as} is total time shift of symbols, u_s is the quantization after periodic sampling time T_s'. The baseband symbol sampling signal is described as below:

$$r_s[n] = r(n \cdot T_s' + u_{as}) = r(n \cdot T_s \cdot (1+\delta) + u_{as})$$
$$= \sum_k X'[k] \cdot \exp\{ j2\pi \frac{k}{N} \cdot (1+\delta) \cdot n\}$$
$$X'[k] = X[k] \cdot \exp\{ j2\pi \frac{k}{N} u_s\} \quad (2)$$

The N means total numbers of carriers; N_g is the length of repeat signal. After demodulation (G_{imp} is the ICI caused by SFO. It can be ignored), we will have Rs (the r_s after FFT calculation):

$$R_s[k] = fft\{r_s[n]\} = X'[k] \cdot I_{k,k} + \sum_{g,g\neq k} X'[g] \cdot I_{g,k}$$
$$\cong X'[k] = X[k] \cdot \exp(j2\pi \frac{k}{N} u_s) \quad (3)$$

Furthermore, u_s is the total time shift with S_{th} symbols because of the periodic sampling frequency offset (caused by δ) . The function is given by:

$$u_s = u_{s-1} + \delta \cdot (N + N_g) \quad (4)$$

Assume the k as continual pilot. The $R_s[k]$ and $R_{s-1}[k]$ only has one phase error. The $\Delta\theta$ represents the phase error between the adjacent symbols. The SFO can be modified by using this relation between these two subcarriers. It can be expressed as follows:

$$\Delta\theta = angle(R_s[k]) - angle(R_{s-1}[k])$$
$$= 2\pi \frac{k}{N}(u_s - u_s - 1) = 2\pi \frac{k}{N} \delta \cdot (N + N_g) \quad (5)$$

After these calculations, it still has the residual subcarriers frequency offset. We have to consider the average phase shift. Finally, we can get the estimation of SFO as below:

$$\hat{\delta} \cong \underset{k}{mean}\{\frac{\Delta\theta_k - \varphi}{2\pi \cdot \frac{k}{N} \delta \cdot (N + N_g)}\}$$

Continual pilot

$$\varphi \cong \underset{k}{mean}\{\Delta\theta_k\} \quad (6)$$

2) Reed-Solomon decoder optimization: As we know, the RS decoder is the most time-consuming part of software DVB-T receiver. Hence, we want to improve the performance of RS decoder [5] which has four parts, and each part uses different algorithms that are described as below: First, there are 4064 additions and 4080 multiplications in GF (256) for getting syndrome:

$$S_0 = r(\alpha^0) = r_0 + r_1\alpha^0 + r_2\alpha^0 + \cdots + r_{254}\alpha^0$$
$$S_1 = r(\alpha^1) = r_0 + r_1\alpha^1 + r_2\alpha^2 + \cdots + r_{254}\alpha^{254}$$
$$S_2 = r(\alpha^2) = r_0 + r_1\alpha^2 + r_2\alpha^4 + \cdots + r_{254}\alpha^{253}$$
$$\vdots$$
$$S_{15} = r(\alpha^{15}) = r_0 + r_1\alpha^{15} + r_2\alpha^{30} + \cdots + r_{254}\alpha^{240}$$

1st vector table | 2nd vector table | 255th vector table

The lookup vector tables will be used to replace these operations. Thus, we create 255 vector tables; each table represents the all possible answers of 16 multiplications in GF (256). You can get 16 multiplications in GF (256) to look up vector table once. The multiplication of GF (256) is unnecessary. Second, we use the GF (256) with lookup vector tables and rewrite the whole Chien search program [8] within the assembly language to replace the multiplications. The performance of the Chien search program will be improved greatly by means of these modifications. It means that the decoding speed is much faster than the original Chien Search program.

TABLE Ⅱ. 2ND VECTOR TABLE

r_1	$r_1\alpha^0$	...	$r_1\alpha^{15}$
1	1	…	38
255	255	…	174

Third, to Reduce FOR Loop and IF Branch: The branch commands like IF, FOR, WHILE, etc., the branch prediction errors will cause the speed of execution down. We need to reduce all the loops in our program by means of using a large number of C or assembly code. Thus, we rewrite some program to generate those C or assembly codes. We write a program to produce a large C code to expand it.

 ISBN: 978-1-61208-154-0

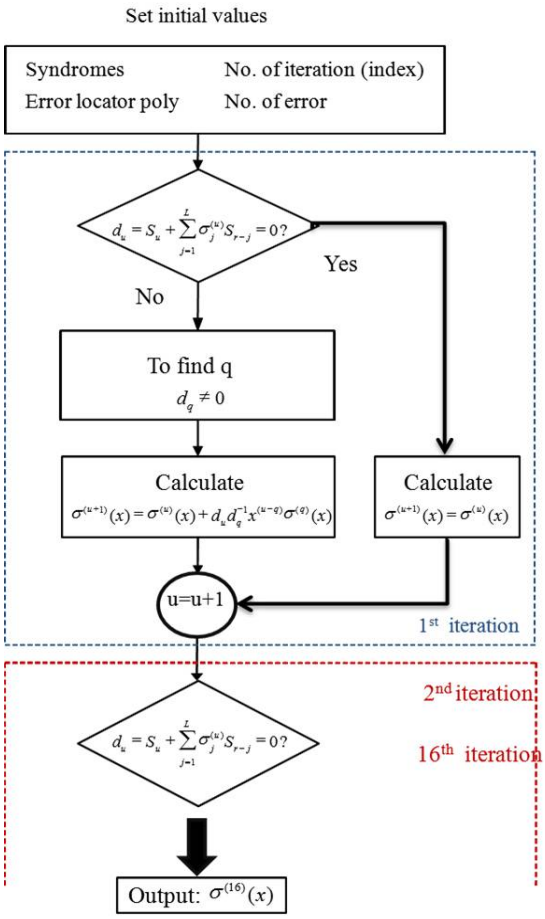

Figure 6. The modified procedure of BM(Berlekamp-Massey) [6] in our program

Besides, the 64-bit CPU provides sixteen XMM registers. It is enough and useful to place all operational values of Forney algorithm procedures by using these registers. It can reduce the slow operation of transferring data between registers and memories as much as possible. The data are only moved among registers.

3) *Viterbi decoder and sytematic approach of software optimization:* Reed-Solomon decoder optimization: In order to speed-up the Viterbi decoder program for software DVB-T receiver. We use the Explorer in Microsoft Visual Studio 2008 (Team Suite edition) to find out and analyze the slowest parts of our programs. Hence, we propose various ways to improve the performance which are described as below [4]: The Viterbi decoder consists of several units: BMU (Branch Metric Unit), ASCU (Add-Compare-Select Unit), PMMU (Path Metric Memory Unit), and SMU (Survivor Memory Unit). The results of the channel estimation also are used to help the decoder to improve the BER (Bit Error Rate). Meanwhile, in order to improve our decoder performance we need some useful optimization methods. The procedures of the optimization are shown in Figure 7.

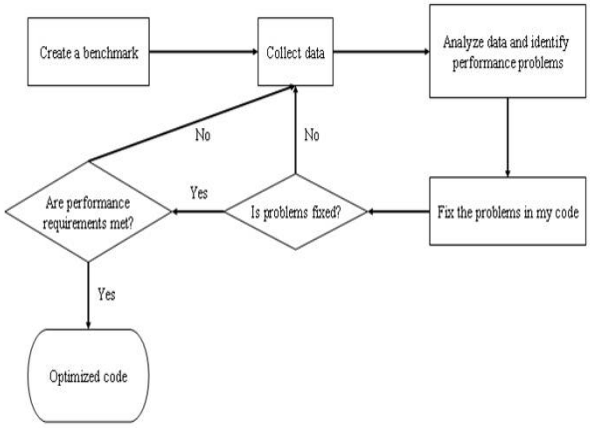

Figure 7. The systematic approach of software optimization.

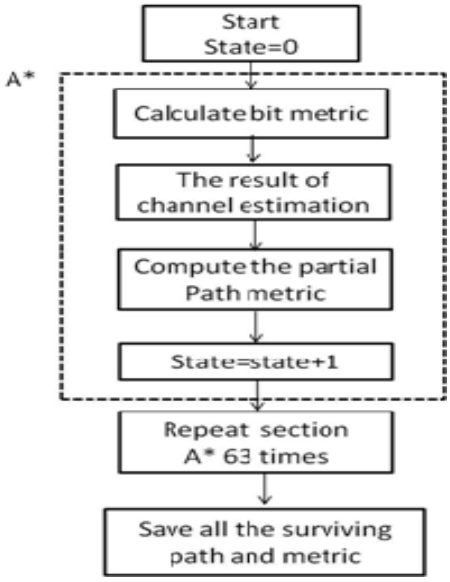

Figure 8. The flowchart of the modified program. A loop is expanded to 64 identical segments A*.

C. SSE4 Instuctions

The newest Intel CPU has the SSE4 (Streaming SIMD Extension 4) instructions [4]. We use the new Intel CPU SIMD instructions to be our tool in order to save the CPU operation time. We can combine it by making some operations of these instructions, and it will be much more efficiency than before.

The execution efficiency of the procedure will drop greatly when containing the judgment operations in the procedure (like IF, FOR, WHILE, etc.).

The efficiency of the procedures would be improved greatly by reducing these operations in the program. General speaking, the FOR loop in C program is used to deal with the continued and similar procedure. In our original Viterbi decoder, the program has many FOR loops. Hence, we create

 ISBN: 978-1-61208-154-0

a lot number of C or assembly code in order to reduce all the loops in our program. (The new procedure is shown in Figure 8).

D. The Other Efficient Methods

The same result can often be got through different ways when writing the program. Therefore, we have to find out or design the new algorithm to speed up the operation of procedures. In fact, there are some other structures that can be used to replace the conventional structures. For example, the C code can be improved by changing the structure. The every 2 coded bits contains only 4 types of bit metric and 2 types of channel estimation results. Therefore, the structure can be modified by combining these computations (shown in Figure 9). The assembly and SSE instructions can implement it.

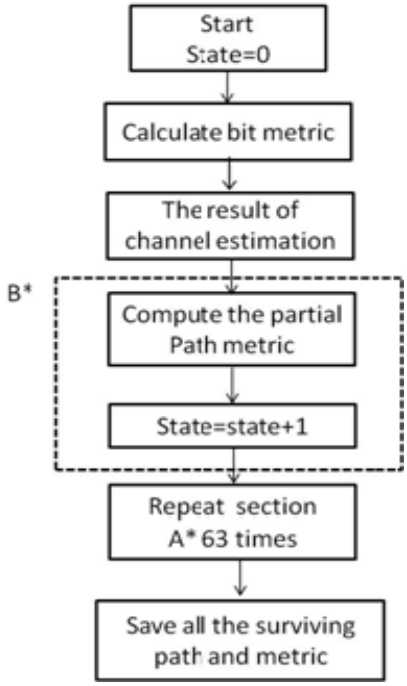

Figure 9. The flowchart of the simplified program. The bit metric calculation and channel estimate multiplication are not repeated 64 times as in Figure 8.

E. Rewrite The Whole Viterbi Coder Program With Assenbly Language

The 64-bit CPU provides more registers to reduce the operation of moving data. The x32 compiler is only allowed to use 8 of 16 XMM registers in the 64 bit CPU. Hence, we have to use all XMM registers by using x64 compiler. Furthermore, we can use the sixteen to reduce the operation time of moving data from registers to memories or moving data from memories to registers. But it is not allowed to use inline assembly by using x64 compiler. The whole programs need to be rewritten within assembly language to adapt to the x64 compiler. The additional eight XMM registers will be used to hold the metric data in order to reduce the movements between memories and registers.

III. IMPLEMENTATION AND RESULTS

The environment of purposed receiver is a desktop PC the specification of which is listed in Table Ⅲ. The OS is windows 7. The software languages are C, C++, C#, or Assembly.

The programming tools are mainly from Microsoft and Intel including compiler and mathematical library.

TABLE Ⅲ. PC PLATFORM SPECIFICATION INCLUDING OS.

Component	Spec.
OS	Windows 7 (64bit)
CPU	Intel(R) Core(TM) i7-2600K CPU @ 3.40GHz (8 CPUs), ~3.4GHz
RAM	4.0 GB
Motherboard	ASUS P8H67-M PRO(REV3.0)

In this research, the name, usage and vender for all tools are listed in Table Ⅳ. The CPU loading of each individual block is listed in Table Ⅴ. Furthermore, The DVB-T signal is also used to input our proposed system. We can get the final results shown in Figure 10. According to the results, it has no interference in this figure.

TABLE Ⅳ. DEVELOPMENT TOOLS.

Tool	Vender	Using for
Visual studio. Net	Microsoft	Compiler, Application
Windows Platform SDK	Microsoft	Driver
DirectShow	Microsoft	API

TABLE Ⅴ. THE CPU LOADING OF EACH BLOCK

Block	Elapsed Inclusive Time (ms)
Time & Frequency Synchronize	63.66
Remove CP & FFT	67.82
Channel estimation	145.87
Deinner & Depuncher	54.49
Deoutter interleave	17.93
Demodulator	8.93
Viterbi decoder	1096.79
RS Decoder	30.67
Descrambler	0.92
Frame Synchronize	8.58
Program Initialization	27.52
Phase Compensation	8.44
C++ standard library	10.27
total	1541

 ISBN: 978-1-61208-154-0

Figure 10. Software Demodulator Result

IV. CONCLUSION

As we know, the real-time bit rate of DVB-T is 9.953 Mbps (16 QAM, guard interval 1/4, rate 2/3 convolutional code, non-hierarchical system for 6 MHz channels) in Taiwan. In this research, we utilize lookup tables, SIMD instructions, loop expansion and the other efficient methods to greatly improve the decoding speed of our Reed-Solomon and Viterbi decoder. The decoding rate of the proposed system is more than the required bit rate of the real-time DVB-T. So, our system is fast enough to process the DVB-T signal used in Taiwan in real-time.

The proposed system is easy to operate with the current commercial PC and it helps us to develop new baseband algorithm. Therefore, we can verify it in real-world environment. It only takes 1541ms to decode the 2760ms video data in our research. Thus, the real-time software DVB-T receiver is possible to be accomplished after these proposed modifications.

REFERENCES

[1] N. Kubo, S. Kondo, and A. Yasuda, “Evaluation of code multipath mitigation using a software GPS receiver,” IEICE Trans. Commun., vol. E88-B, no. 11, pp. 4204-4211, Nov. 2005.

[2] Shu-Ming Tseng, Yueh-Teng Hsu, Meng-Chou Chang, and Hsiao-Lung CHAN, “A Notebook PC Based Real-Time Software Radio DAB Receiver, ” IEICE Trans. Commun., vol. E89-B, no. 12, pp. 3208-3214, Dec. 2006.

[3] ETSI EN 300 744 :Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for digital terrestrial television ETSI, 2004.

[4] Shu-Ming Tseng,Yu-Chin Kuo,Yen-Chih Ku, and Yueh-Teng Hsu, “Software Viterbi Decoder with SSE4 Parallel Processing Instructions for Software DVB-T Receiver, ” in Proc. The 7th IEEE International Symposium on Parallel and Distributed Processing with Applications (ISPA-09), Aug. 2009, pp. 102-105.

[5] Shu-Ming Tseng, Jian-Cheng Yu, Jheng-Zong Shih, and Yueh-Teng Hsu, “Reed-Solomon Decoder Optimization for PC-Based DVB-T Software Radio Receiver, ” in Proc. The International Conference on Consumer electronics, Jan. 2011, pp. 393-394.

[6] Michael Speth, Stefan Fechtel, Gunnar Fock, and Heinrich Meyr, “Optimum Receiver Design for OFDM-Based Broadband Transmission—Part II: A Case Study, ” IEEE Trans. Commun., vol. 49, no. 4, pp. 571-578, Apr. 2001.

[7] Michael Speth, Stefan Fechtel, Gunnar Fock, and Heinrich Meyr, “Optimum Receiver Design for Wireless Broad-Band Systems Using OFDM—Part I: A Case Study, ” IEEE Trans. Commun., vol. 47, no. 11, pp. 1668-1677, Nov. 1999.

[8] S. B. Wicker, Error Control Systems for Digital Communications and Strorage, Prentice Hall, 1995.

 ISBN: 978-1-61208-154-0