*Send Orders for Reprints to reprints@benthamscience.net*

*The Open Electrical & Electronic Engineering Journal,* **2014,** *8,* **19-26**



Open Access

# Web-based HW/SW Codesign Automation of an 802.16 Channel Codec

Ching-Shun Chen[*], Chi-Yang Liao[#], Chih-Jen Hsiao[#], Yuan-Wen Huang[#], Hsing-Chi Fang[#], Li-Shin Huang[#] and Hsi-Ju Chen[#]

*Department of Electrical Engineering, Da-Yeh University, 168 University Rd, Dacun, Changhua, Taiwan 51591, ROC*

**Abstract:** The intelligent web-based systems had been developed for various applications such as conceptual design, rapid prototyping manufacturing, and supply chain planning. On the other hand, hardward/software (HW/SW) codesign is commonly used methodology for digital-system design. Noteworthy, our previous work had carried out in order to develop a web-based HW/SW codesign framework with implementation of a JPEG encoder [1]. Design of an 802.16 codec with the web-based HW/SW codesign automation is further performed in this work. The 802.16 codec designed by C program language is compiled with MS Visual C++ 6 to verify its correctness at first. The ANSI C design of an 802.16 codec is further transferred into the developed web-based codesign framework, which utilize web servers, CGI programs (in C or PERL), GCC compiler, SPIM simulator, and Cadence EDA tools. In about seventy working minutes, the respective design of a MIPS-like processor (hardware) in TSMC 0.18 μm VLSI layout with machine code (software) is going to be automatically accomplished in the web-based codesign framework. The full detailed HW/SW design is capable to be automatically e-mailed return to remote user for further application. In the web-based codesign framework, the MIPS-like processor is designed by using alogrithmic state machine and Verilog hardware description language. In addition, the HW/SW design of an 802.16 codec is verified with using a Xilinx FPGA development board for this study.

**Keywords:** 802.16, HW/SW codesign automation, MIPS, FPGA, EDA, VLSI.

## 1. INTRODUCTION

The intelligent web-based systems had been developed for conceptual design, rapid prototyping manufacturing and supply chain planning [2-4]. As inter-enterprise cooperation on the internet becomes more dynamic and changing, the concept of how internet-enabled designs may allow companies to create global design groups to complete complex system-on-chip designs [5]. The HW/SW codesign methodology is commonly used for digital-system design, especially it is aimed at the consumer market to execute a variety of high-performance real-time tasks, such as audio, images, video, radio channel access, TCP/IP protocol stack management and so on. Noteworthy, the internet-based collaborative HW/SW codesign framework has been developed in our previous study with very preliminary verification with the implementation of an JPEG encoder [1]. To validate the performance of developed framework, more advanced and complicated design is urgently required to be studied.

WiMAX, meaning Worldwide Interoperability for Microwave Access, is based on the IEEE 802.16 standard [6] (also called Broadband Wireless Access). WiMAX has been tipped to bring a revolution in the way we use broadband services. In this study, design of the 802.16 channel codec capable for the application of WiMAX is performed in association with the web-based HW/SW codesign automation. The web-based HW/SW codesign framework which utilizes web servers, CGI programs [7] (in C or PERL), GCC compiler, SPIM simulator, and various EDA tools such as Cadence BuildGates and SOC Encounter has been implemented for running in Windows XP and TopologiLinux. As the processor core for HW/SW codesign, the MIPS-like processor is designed by using algorithmic state machine [8] and Verilog hardware description language.

An 802.16 channel codec designed by C program language is compiled with MS Visual C++ 6 to verify its correctness at first. Furthermore, a remote user can input the ANSI C design of the 802.16 channel codec to the web-based HW/SW codesign framework. In about seventy minutes, the full detailed HW/SW designs including C original design, assembly code, machine code and a MIPS-like processor in TSMC 0.18 μm VLSI layout are going to be automatically accomplished and e-mailed return to the remote user for further application. The HW/SW design of an 802.16 channel codec is verified by using MS Visual C++ 6, PCSPIM, Modelsim and a Xilinx FPGA development board [9] in this work.

The objective of this work is to further validate our novel web-based HW/SW codesign framework [1] which can effectively integrate various EDA tools to complete an 802.16 HW/SW codesign in less than two working hours. Using the proposed codesign automation is anticipated to save dozens or hundreds times of labor efforts than those using the EDA tools separately without design automation.

*Address correspondence to this author at the Department of Electrical Engineering, Da-Yeh University, 168 University Rd, Dacun, Changhua, Taiwan 51591, ROC; Tel: +886-4-8511888; Ext.: 2175;
Fax: +886-4-8511245;
E-mail: cschen@mail.dyu.edu.tw
[#]Graduate Student

1874-1290/14 

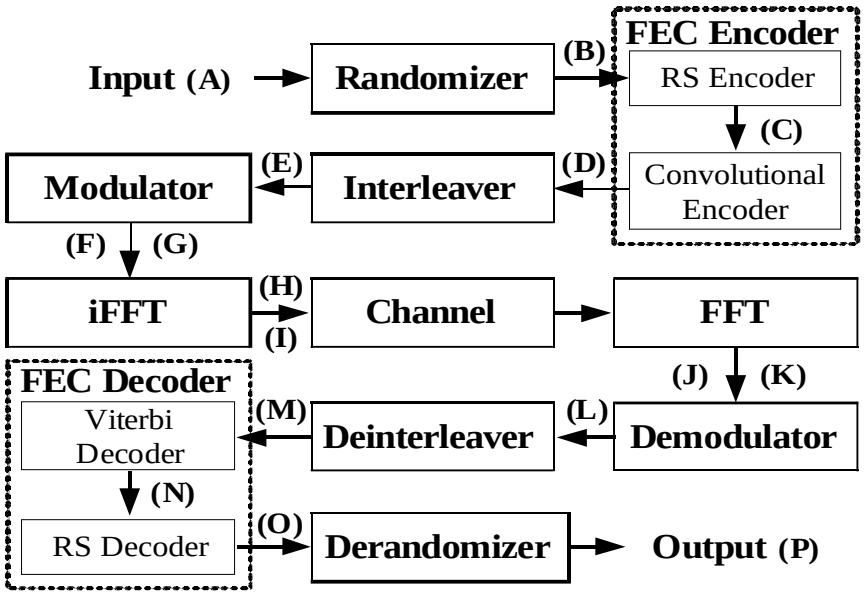


**Fig. (1).** The flowchart of an 802.16 channel codec.

The paper is organized as follows: Section 2 introduces an 802.16 channel codec; The development of a MIPS-like processor using ASM and Verilog is discussed in Section 3; Our approach to realize the web-based HW/SW codesign framework and various possibilities is described in Section 4; Finally, Section 5 presents the various verification results to show correctness of the design of an 802.16 channel codec with web-based HW/SW codesign automation.

## 2. 802.16 CHANNEL CODEC

Fig. (**1**) presents the flowchart of an 802.16 channel codec [6] and indicates that symbols, (A)-(P), are assigned to be referred to the following verification results. Randomizer is the first process after receiving the input signals. The Forward Error Correction (FEC) encoding is performed by passing the data in block format through the Reed-Solomon (RS) encoder and then passing it through a convolutional encoder. After interleaving, the signals are entered serially to modulate. The inverse fast Fourier transform (iFFT) is subsequently to convert the frequency domain signals into time domain. The time domain signals are transmitted through the channel. After channel transmission, the received signal is processed by using the fast Fourier transform (FFT) and demodulator. Furthermore, the signals are restored back into the original form by deinterleaver, FEC decoding and derandomizer. In general, the original data signal will be recovered in the final output.

In the 802.16 channel codec, the randomization of the data signal is performed to avoid long sequences of consecutive ones and zeros. The randomizer is implemented with a Pseudo Random Binary Sequence generator using an XOR operation with a generator polynomial of $1+X^{14}+X^{15}$. Forward Error Correction (FEC) is performed on two phases through the Reed-Solomon (RS) outer code and the convolutional inner code to correct burst error. The RS code corrects burst error at the byte level. The convolutional code corrects independent bit errors.

In IEEE 802.16 standard, interleaving defined by a two-step process is used against long sequences of consecutive errors. The first step ensures that adjacent coded bits are mapped onto non-adjacent subcarriers. The second step ensures that adjacent coded bits are mapped alternately onto less or more significant bits of the constellation, thus avoiding long runs of unreliable bits. The specified modulation scheme can be BPSK, QPSK, 16-QAM and 64-QAM in the 802.16 channel codec. The BPSK and QPSK modulations are used for communication over longer distances or in impaired conditions. On the other hand, the disadvantage of BPSK and QPSK lies in the low signal rate. In this work, the QPSK modulation is performed.

## 3. MIPS-LIKE PROCESSOR

As well known in open literatures, MIPS processor was originally invented as part of a Stanford research project and later brought to market by newly-started MIPS Corporation in 1985, releasing the MIPS R2000 [10] running at 8 MHz on 2.0 micron process. MIPS, a family of 32-bit and 64-bit computer processors, is currently primarily used in many embedded applications such as the Windows CE devices, network routers, and video game consoles like the Nintendo 64, PlayStation 2, and PlayStation Portable handheld system. Until late 2006 they were also used in many of SGI's computer products. In this work, the MIPS-like processor architecture, which can be generated automatically by web-based HW/SW codesign framework, includes thirty-two general-purpose 32-bit registers, forty-seven instructions, and an external floating-point unit (FPU) for computation of real numbers.

For the design of MIPS-like processor, Algorithmic State Machine (ASM) notation [8], which builds on the foundation of state transition diagrams by providing a mean to express the state trajectory as a conditional flow, is applied in this work. It is accomplished by incorporating conditions, deci-

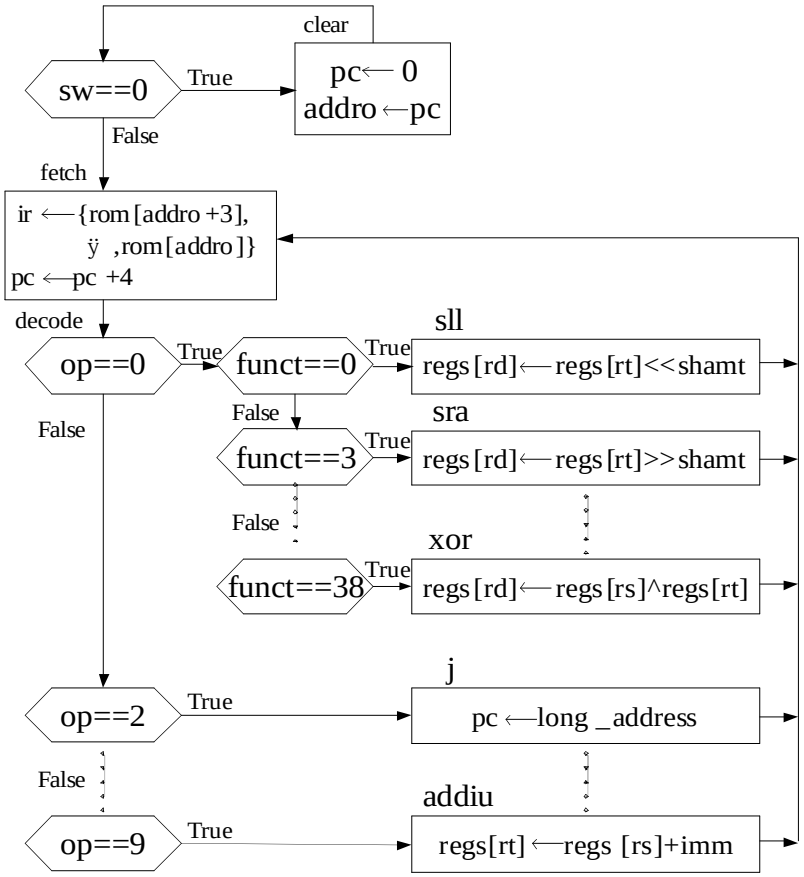


**Fig. (2).** A simplified ASM chart for MIPS-like processor.

sion points, and distinct output points within the ASM charts. ASM charts are composed of multiple ASM blocks, each block with exactly one state, at least one output, and exit conditions. In the ASM chart, each state takes an exact amount of time, as known as the clock period or clock cycle. ASM blocks have a single entry point, but can have multiple exit paths. Fig. (**2**) shows a simplified ASM chart for the designed MIPS-like processor.

Verilog is a hardware description language (HDL) utilized to model electronic systems in VLSI design industry. Based upon the designed ASM chart, the MIPS-like processor can be behaviorally modeled with Verilog very efficiently. The behavioral Verilog model of MIPS-like processor could be simulated and synthesized for further verification. In addition, EDA tools such as Cadence BuildGates and SOC Encounter are applied to accomplish VLSI layout of MIPS-like processor.

## 4. WEB-BASED HW/SW CODESIGN

Fig. (**3a**) presents the realized architecture for the web-based codesign framework. In the developed web-based HW/SW codesign framework, Windows XP and Topologil-inux operating systems are installed in one computer (CPU 2.0 GHz, RAM 752 MB, HD 30 GB) in association with Abyss web server, CGI programs, GCC compiler, SPIM simulator, and Cadence BuildGates and SOC Encounter.

In this work, web server provides access of the user to the codesign framework, as well as processes all design requests received from users who use their web browsers for communication with the server. Another function of the web server is to provide information about the current design status and results of the framework (i.e. the status and results of actions requested by the user). As the matter of fact, the web server is a connecting link between the hardware of the internet network and the user's browser. The CGI programs designed by using C or Perl language are utilized to make dynamic template pages (written in HTML language), to grant or deny user access by checking user's IP address, and to drive GCC compiler, SPIM simulator, Cadence BuildGates, and SOC Encounter doing their jobs, respectively. Another function implemented in CGI programs is the organization of the files with result of simulation task and with VLSI layout of the designed MIPS-like processor. The processed design results including C original design, assembly code, machine code, and VLSI layout with timing report, area report, and power report may be sent to client *via* e-mail when the full design task is accomplished. Fig. (**3b**) illustrates a basic demonstration for the web-based HW/SW codesign.

A website developed for internal use and organized as aforementioned discussing offers the following possibilities:

I. Users can develop their designs in various fields such as wireless communication, digital signal processing, and

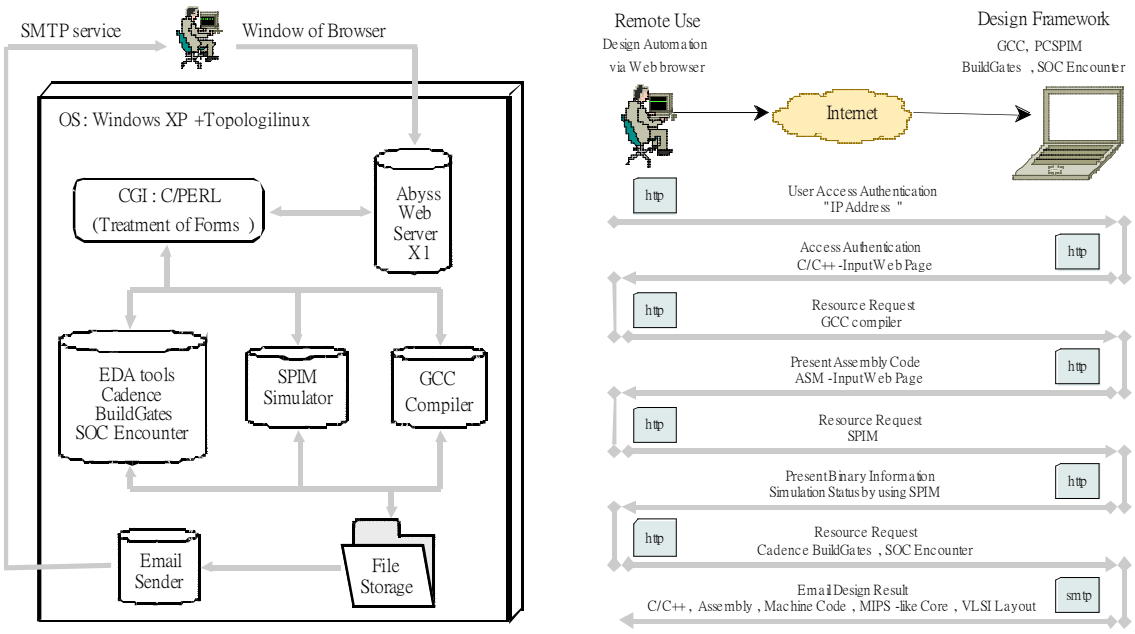


**Fig. (3).** (**a**) Block diagram of the web-based design framework. (**b**) Demonstrator operation, and layout example.

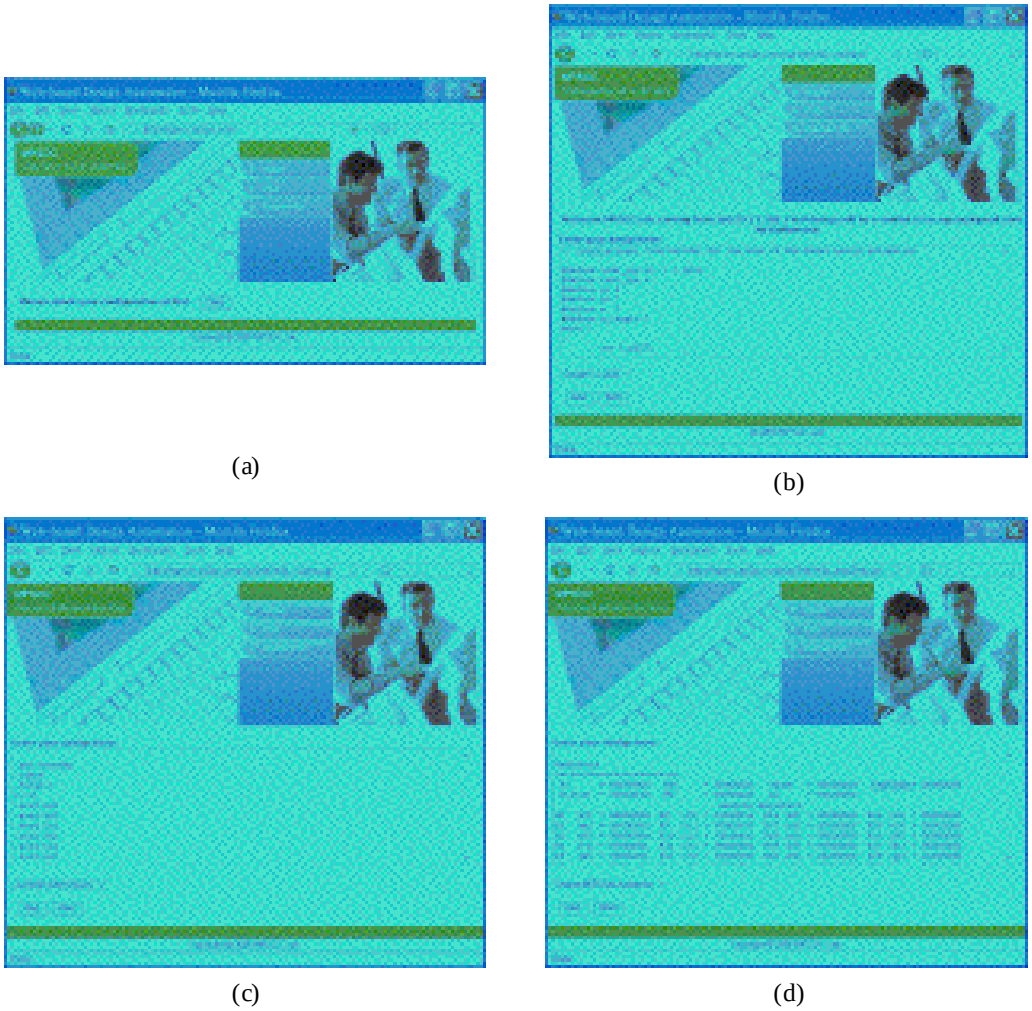


**Fig. (4).** (**a**) Main page of web-based design framework. (**b**) The example of user permission and C/C++ design input. (**c**) The example of assembly code generated by GCC compiler. (**d**) The example of simulation completed by SPIM simulator.

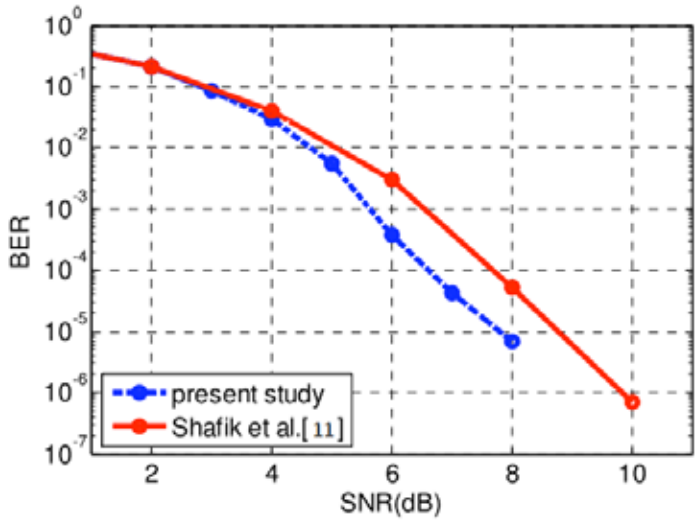


**Fig. (5).** Performance of the 802.16 codec in AWGN channel.

| Symbol | Description |
|---|---|
| (A) | input data |
| (B) | randomized data |
| (C) | Reed –Solomon encoded data |
| (D) | convolutionally encoded data |
| (E) | interleaved data |
| (F) | real part of the modulated data |
| (G) | imaginary part of the modulated data |
| (H) | real part of the converted data by using iFFT |
| (I) | imaginary part of the converted data by using iFFT |
| (J) | real part of the converted data by using FFT |
| (K) | imaginary part of the converted data by using FFT |
| (L) | demodulated data |
| (M) | deinterleaved data |
| (N) | Viterbi decoded data |
| (O) | Reed–Solomon decoded data |
| (P) | derandomized / output data |

(a)

| | |
|---|---|
| (A) | 0 1 0 0 0 1 0 0 |
| (B) | 1 1 0 1 0 1 0 1 |
| (C) | 213 210 60 46 23 36 134 93 |
| (D) | 1 1 1 0 1 0 1 1 |
| (E) | 1 1 1 0 1 0 1 1 |
| (F) | -0.707107 -0.707107 -0.707107 -0.707107<br>0.707107 0 .707107 -0.707107 0 .707107 |
| (G) | 0.707107 -0.707107 -0.707107 0 .707107<br>0.707107 -0.707107 -0.707107 0 .707107 |
| (H) | 0.405330 -1.169104 0 .755880 -0.014319<br>1.021734 0 .150323 -0.742340 -0.238936 |
| (I) | -0.390165 -0.509207 1 .182657 -0.033251<br>-0.447060 -0.688592 -0.402285 0 .145099 |
| (J) | -0.231133 -0.255985 0 .267924 0 .136301<br>0.091419 0 .119443 -0.054094 0 .125970 |
| (K) | 0.095426 -0.028831 0 .162254 -0.284182<br>-0.270931 -0.278726 0 .219232 -0.412230 |
| (L) | 1 1 1 0 1 0 1 1 |
| (M) | 1 1 1 0 1 0 1 1 |
| (N) | 1 1 0 1 0 1 0 1 |
| (O) | 213 210 60 46 23 36 134 93 |
| (P) | 0 1 0 0 0 1 0 0 |

(b)

**Fig. (6).** (**a**) Every subsection of the outputs from each process with respect to Fig. (**1**). (**b**) Partial results extracted from MS VC++6.

information technology by using popular C/C++ languages. The custom C/C++ codes can be further compiled, assembled, and embedded into whole HW/SW codesign *via* the window of browser just in hours.

II. Users can use the email client program to receive whole design including C/C++ original design, assembly code, machine code, and VLSI layout with timing report, area report, and power report. Fig. (**4**) illustrates some possibilities of the studied framework for HW/SW codesign *via* internet.

## 5. VERIFICATION

The performance simulation of the studied 802.16 channel codec is shown in Fig. (**5**) with comparison to that of previous study by Shafik *et al.* [11]. It is found that performance trends of this work and previous study are similar, but a little difference in BER performance is due to Shafik *et al.* using convolutional code only without RS code in their simulation.

The input signals of an 802.16 channel codec are given by {0, 1, 0, 0, 0, 1, 0, 0, ...} to verify the correctness of whole design. For the design verification, (A)-(P), are applied to illustrate the outputs from each component of an 802.16 channel codec as shown in Fig. (**6a**) with respect to Fig. (**1**).

The 802.16 channel codec designed by C program language is compiled with MS Visual C++ 6 to verify its correctness. Partial result data is extracted for further verification as shown in Fig. (**6b**). The two red frames of Fig. (**6b**) show that the original input data, subsection (A), is the same as the restored output data, subsection (P).

```
(A) [0x10010050 ] 0x00000000 0 x00000001 0 x00000000 0 x00000000
    [0x10010060 ] 0x00000000 0 x00000001 0 x00000000 0 x00000000
(B) [0x10011340 ] 0x00000001 0 x00000001 0 x00000000 0 x00000001
    [0x10011350 ] 0x00000000 0 x00000001 0 x00000000 0 x00000001
(C) [0x10010848 ] 0x000000 d5 0x 000000 d2 0 x0000003 c 0x0000002 e
    [0x10010858 ] 0x00000017 0 x00000024 0 x00000086 0 x0000005 d
(D) [0x100113 e0] 0x00000001 0 x00000001 0 x00000001 0 x00000000
    [0x100113f0] 0x00000001 0 x00000000 0 x00000001 0 x00000001
(E) [0x100119 e4] 0x00000001 0 x00000001 0 x00000001 0 x00000000
    [0x100119f4] 0x00000001 0 x00000000 0 x00000001 0 x00000001
(F) [0x10012bf0] 0xbf3504f3 0xbf3504f3 0xbf3504f3 0xbf3504f3
    [0x10012c00] 0x3f3504f3 0x3f3504f3 0xbf3504f3 0x3f3504f3
(G) [0x100131f4] 0x3f3504f3 0xbf3504f3 0xbf3504f3 0x3f3504f3
    [0x10013204 ] 0x3f3504f3 0xbf3504f3 0xbf3504f3 0x3f3504f3
(H) [0x10014048 ] 0x3ecf876 f 0xbf 95 a533 0 x3f418155 0 xbc 6a9ba0
    [0x10014058 ] 0x3f82c82f 0x3e19ee40 0 xbf3e09 ff 0xbe74ab 94
(I) [0x1001454c] 0xbec 7c3bd 0xbf025 b69 0 x3f97614 c 0xbd 0831 c4
    [0x1001455c] 0xbee 4e516 0 xbf 304799 0 xbecdf 840 0 x3e1494 ce
(J) [0x10011cec] 0xbe 6cae42 0 xbe 831064 0 x3e892 d4c 0x3e0b925a
    [0x10011cfc] 0x3dbb 39 fc 0x3df49e6c 0xbd 5d9188 0 x3e00fe30
(K) [0x100120f0] 0x3dc 36ec 4 0xbcec 2f60 0 x3e2625 f8 0xbe 918051
    [0x10012100 ] 0xbe 8ab776 0 xbe 8eb528 0 x3e607 e68 0 xbed 30fd9
(L) [0x10013f10] 0x00000001 0 x00000001 0 x00000001 0 x00000000
    [0x10013f20] 0x00000001 0 x00000000 0 x00000001 0 x00000001
(M) [0x10014514 ] 0x00000001 0 x00000001 0 x00000001 0 x00000000
    [0x10014524 ] 0x00000001 0 x00000000 0 x00000001 0 x00000001
(N) [0x1001501c] 0x00000001 0 x00000001 0 x00000000 0 x00000001
    [0x1001502c] 0x00000000 0 x00000001 0 x00000000 0 x00000001
(O) [0x100112 e4] 0x000000 d5 0x 000000 d2 0 x0000003 c 0x0000002 e
    [0x100112f4] 0x00000017 0 x00000024 0 x00000086 0 x0000005 d
(P) [0x10015984 ] 0x00000000 0 x00000001 0 x00000000 0 x00000000
    [0x10015994 ] 0x00000000 0 x00000001 0 x00000000 0 x00000000
```

**Fig. (7).** Partial results extracted from PCSPIM.

The C design of the 802.16 channel codec is further compiled by using GCC compiler toolchain and assembled by using SPIM to generate the MIPS-architecture assembly program and to complete MIPS-machine simulation, respectively, within the web-based HW/SW codesign framework. Partial coding data extracted from PCSPIM is shown in Fig. (**7**). It is found that data contents from using MS VC++6 are the same as those from using PCSPIM. For instance, the first data of subsection (C) in Fig. (**6b**) is decimal 213 and is the same as hexadecimal 0x000000d5 corresponding to address 0x10010848 in Fig. (**7**). The first data 0x3ecf876f corresponding to address 0x10014048 in subsection (H) of Fig. (**7**) can be converted to 0.40533015 with IEEE 754 standard and is almost the same as the first data 0.405330 of subsection (H) in Fig. (**6b**).

The full detailed design resulted from web-based HW/SW codesign framework includes the Verilog behavioral models of a MIPS-like processor and an 802.16 channel codec. The Verilog model of a MIPS-like processor embedded with an 802.16 channel codec can be simulated by using the EDA tool, Modelsim. The simulated results from Modelsim are the same as those from MS VC++6 and PCSPIM. Partial memory contents extracted from Modelsim as shown in Fig. (**8**) are the same as those in Fig. (**6b**) and (**7**). For example, the data 0x3ecf876f corresponding to address 0x10014048 in subsection (H) of Fig. (**7**) is the same as reverse ordered data 6f 87 cf 3e corresponding to address 4048 in subsection (H) of Fig. (**8**).

The Virtex-II PRO(V2-Pro) FPGA development system interfacing with flash memory is applied to verify the aforementioned Verilog model again. The behavioral Verilog model is synthesized and programmed into the FGPA board by using ISE design suite. The original data and the 802.16 channel codec program are programmed into flash memory, and the synthesized Verilog model of the MIPS-like processor is programmed into FPGA chip, respectively.

When the 802.16 channel coding and decoding are accomplished, the coded and decoded data of an 802.16 channel codec will be appeared in flash memory finally. The partial coding and decoding results in flash memory are the same as those by using MS VC++6 (Fig. **6**), PCSPIM (Fig. **7**), and Modelsim (Fig. **8**). The whole 802.16 channel codec data is extracted from flash memory and is found to be correct in this work.

```
(A)  50 : 00 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00
     60 : 00 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00
(B) 1340: 01 00 00 00 01 00 00 00 00 00 00 00 01 00 00 00
    1350: 00 00 00 00 01 00 00 00 00 00 00 00 01 00 00 00
(C) 848: d5 00 00 00  d2 00 00 00 3  c 00 00 00 2  e 00 00 00
    858: 17 00 00 00 24 00 00 00 86 00 00 00 5      d 00 00 00
(D) 13e0: 01 00 00 00 01 00 00 00 01 00 00 00 00 00 00 00
    13f0: 01 00 00 00 00 00 00 00 01 00 00 00 01 00 00 00
(E) 19e4: 01 00 00 00 01 00 00 00 01 00 00 00 00 00 00 00
    19f4: 01 00 00 00 00 00 00 00 01 00 00 00 01 00 00 00
(F) 2bf0: f3 04 35 bf f3 04 35 bf f3 04 35 bf f3 04 35 bf
    2c00: f3 04 35 3f f3 04 35 3f f3 04 35 bf f3 04 35 3f
(G) 31f4: f3 04 35 3f f3 04 35 bf f3 04 35 bf f3 04 35 3f
    3204: f3 04 35 3f f3 04 35 bf f3 04 35 bf f3 04 35 3f
(H) 4048: 6f 87 cf 3e 33 a5 95 bf 55 81 41 3 f a0 9b 6a bc
    4058: 2f c8 82 3 f 40 ee 19 3e ff 09 3e bf 94 ab 74 be
(I) 454c: bd c3 c7 be 69 5b 02 bf 4c 61 97 3 f c4 31 08 bd
    455c: 16 e5 e4 be 99 47 30 bf 40 f8 cd be ce 94 14 3 e
(J) 1cec: 42 ae 6c be 64 10 83 be 4c 2d 89 3 e 5a 92 0 b 3e
    1cfc: fc 39 bb 3d 6c 9e f4 3d 88 91 5 d bd 30 fe 00 3 e
(K) 20f0: c4 6e c3 3d 60 2 f ec bc f8 25 26 3 e 51 80 91 be
    2100: 76 b7 8 a be 28 b5 8 e be 68 7 e 60 3 e d9 0 f d3 be
(L) 3f10: 01 00 00 00 01 00 00 00 01 00 00 00 00 00 00 00
    3f20: 01 00 00 00 00 00 00 00 01 00 00 00 01 00 00 00
(M) 4514: 01 00 00 00 01 00 00 00 01 00 00 00 00 00 00 00
    4524: 01 00 00 00 00 00 00 00 01 00 00 00 01 00 00 00
(N) 501c: 01 00 00 00 01 00 00 00 0     0 00 00 00 0  1 00 00 00
    502c: 00 00 00 00 0  1 00 00 00 0  0 00 00 00 01 00 00 00
(O) 12e4: d5 00 00 00  d2 00 00 00 3  c 00 00 00 2  e 00 00 00
    12f4: 17 00 00 00 24 00 00 00 86 00 00 00 5      d 00 00 00
(P) 5984: 00 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00
    5994: 00 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00
```

**Fig. (8).** Partial results extracted from Modelsim.

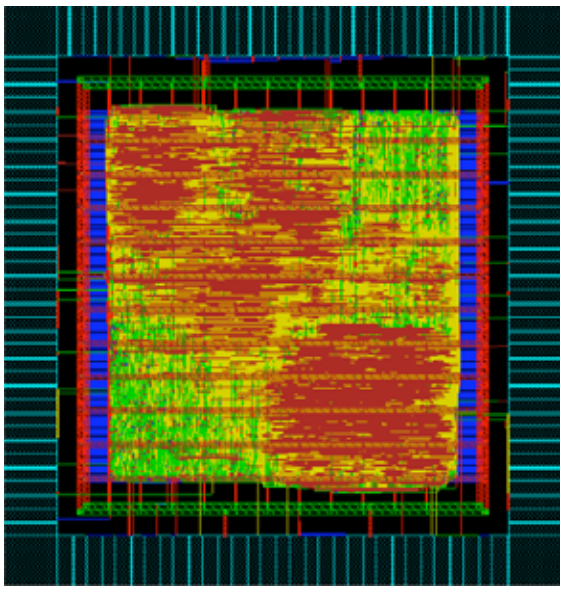

**Fig. (9).** VLSI layout of the MIPS-like processor.

In addition, the Verilog structural model of a MIPS-like processor can be automatically generated by using Cadence BuildGates in web-based HW/SW codesign framework. The synthesized netlist from BuildGates is further imported to Cadence SOC Encounter for VLSI layout with using TSMC 0.18 μm standard cell library as shown in Fig. (**9**). The VLSI layout could be passed for Design Rules Check (DRC) to complete all verification tasks.

## CONCLUSION

Verification results reveal that the design of an 802.16 channel codec by using the web-based HW/SW codesign

automation is correct. The main contribution of this work is to design an 802.16 channel codec to successfully verify a web-based HW/SW framework [1] which can design a MIPS-like processor core and integrate various EDA tools to achieve web-based compilation, assembly, simulation, synthesis, and VLSI layout. In general, a HW/SW codesign for the 802.16 channel codec is able to be accomplished by using the developed framework in less than two working hours. In addition, users can develop their design in C/C++ to make their research very efficiently. Noteworthy, more complicated algorithms need to be studied for the robustness of proposed web-based HW/SW codesign framework.

## CONFLICT OF INTEREST

The authors confirm that this article content has no conflict of interest.

## ACKNOWLEDGEMENTS

The supports from Chip Implementation Center (CIC) and various EDA vendors such as Cadence, Mentor Graphic, and Xilinx are sincerely acknowledged for this work.

## REFERENCES

[1] C.S. Chen, H.C. Fong, C.J. Hsiao, C.Y. Liao, and H.J. Chen, "Design of a JPEG encoder with the web-based HW/SW codesign automation," *Journal of Internet Technology*, vol. 13, pp. 543-550, July 2012.

[2] S.F. Qin, R. Harrison, A.A. West, I.N. Jordanov, and D.K. Wright, "A framework of web-based conceptual design," *Computers in Industry*, vol. 50, pp. 153-164, Feb. 2003.

[3] R.C. Luo and J.H. Tzou, "The development of an intelligent web-based rapid prototyping manufacturing system," *IEEE Transactions on Automation Science and Engineering*, vol. 1, pp. 4-13, July 2004.

[4] C.D. Tarantilis, C.T. Kiranoudis, and N.D. Theodorakopoulos, "A web-based ERP system for business services and supply chain management: Application to real-world process scheduling," *European Journal of Operational Research*, vol. 187, pp. 1310-1326, June 2008.

[5] A. Sangiovanni-Vincentelli, Ed., *The internet: the next IC design environment, in DAC'2000 conference*, Keynote presentation, 2000.

[6] IEEE Standard 802.16-2004, *Air Interface for Fixed and Mobile Broadband Wireless Access Systems*, IEEE, 2004.

[7] Thomas Boutell, *CGI Programming in C and Perl*, Addison-Wesley Professional, 1996.

[8] M.G. Arnold, *Verilog Digital Computer Design: Algorithms into Hardware*, Prentice-Hall, Inc., 1999.

[9] "Hardware reference manual of the Virtex-II PRO development system," http://www.digilentinc.com/Data/Products/XUPV2P/XUPV2P_User_Guide.pdf

[10] D.A. Patterson, and J.L. Hennessy, *Computer Organization and Design: The Hardware/Software Interface, 3rd ed.*, Morgan Kaufmann, 2007.

[11] M. Shafik, H.M. El Badawy, and G. Selim, Eds., *Simulation and Assessment of WiMAX Network Performance In Different Environments, Proceedings of the third IEEE/IFIP International Conference in Central Asia on Internet (ICI)*, pp. 1-5, Sept. 2007.

Received: October 09, 2013 Revised: March 22, 2014 Accepted: March 28, 2014