DSP Curriculum Development for Computer Engineering using Altera's DE2 FPGA Kits

Yahong Rosa Zheng, Sarat K. Chitneni, Daryl G. Beetner

Electrical and computer engineering
University of Missouri-Rolla
Email: *{zhengyr, scmt9, daryl}@umr.edu*

Abstract

This paper presents laboratory materials on Digital Signal Processing (DSP) for Computer Engineering (CmpE) curriculum using Altera's DE2 FPGA (Field Programmable Array) university kits. The Altera's DE2 kit has been adopted by many universities for courses on digital logic, embedded systems, and computer architecture. However, it has not been fully exploited for DSP. We take advantage of the multiplier blocks and the IP (Intellectual Property) cores available in the on-board FPGA to demonstrate the basic digital filter operations. Although the number of multiplier blocks in the DE2 board is limited, we have successfully implemented high pass, low pass, and band pass filters. We also utilize the on-board audio codec to demonstrate real-time data acquisition and processing. By comparing the sound effects of input and output audio signals, the lab exercises demonstrate basic DSP theory such as analog-to-digital converter, frequency analysis, digital filter, and digital-to-analog converter. The designed experiments can be used either in an upper undergraduate level electronics or embedded systems course or in a stand-alone DSP course for both EE and CmpE. It offers excellent integration of the DSP component with other components of the CmpE curriculum and achieves necessary coverage in the limited number of hours allocated.

Introduction

Digital Signal Processing (DSP) courses have been well developed for electrical engineering (EE) curriculum with heavy treatment on theoretical foundations, significant amount of Matlab simulations, and reinforcement by hardware laboratories. The hardware labs are often designed using specialized digital signal processor kits (DSK) such as C54xx/C67xx DSK from Texas Instruments or Blackfin kit from Analog Devices.

Recently, DSP is also recommended as a critical, necessary component for computer engineering (CmpE) curriculum. The importance of DSP design and its integration in to the undergraduate computer engineering curriculum is considered critical for educating the 21st century computer engineers and preparing them to be on par with the ever increasing contribution of DSP in many real-world applications today. However, incorporating DSP into CmpE curriculum presents two major challenges: the limited number of lecture hours allocated to the DSP component and the integration of the DSP component with other CmpE components. The curriculum guidelines [1] suggests only 17 core hours to cover a breadth of DSP history, theories, and concepts, including spectra analysis, discrete Fourier transforms, sampling and digital filters. It also suggests some elective hours to cover discrete time signals, window functions, convolution, and audio/image processing. In comparison, these similar topics are covered in depth in EE curriculum with 63 core hours. Therefore, the courses designed for EE curriculum may not be directly used for CmpE curriculum. Meanwhile, many universities, including the University of Missouri-Rolla, have adopted Field Programmable Gate Array (FPGA) for many core CmpE courses [6] such as digital logic, embedded systems, microprocessors, and computer architecture. The DSP on FPGA approach [2-5] becomes more attractive to implementing DSP component for CmpE curriculum than digital signal processors traditionally used for the EE curriculum. Besides, the increasingly important role played by FPGA in the DSP market also gives strong justification to the adoption of FPGA for our DSP laboratories.

This paper presents our new development on DSP laboratory materials for CmpE curriculum using Altera's DE2 FPGA kits. Many universities have developed DSP labs based on more sophisticated FPGAs such as Xilinx Vertex 4 and Altera's Stratix II. However, low-end FPGA such as the Cyclone II on Altera's DE2 board have mainly been used as simple input/output (I/O) and micro-controller. Teaching materials are found only for digital logic and computer architecture courses in both the CDROM provided by Altera and university course websites. To the best of our knowledge, there is no DSP lab material based on DE2 FPGA board available online at the time of this writing.

In this work, we take advantage of the multiplier blocks and the IP (Intellectual Property) cores available in the on-board FPGA to demonstrate the basic digital filter operations. Although the number of multiplier blocks in DE2 boards is limited and basic logic elements have to be used to implement the accumulator to complete a Multiply-Accumulate (MAC) operation, we have successfully implemented high pass, low pass, and band pass filters. We have also utilized the on-board audio codec to implement real-time data acquisition and processing. The input and output audio signals are compared to demonstrate the functionalities of the filters as well as the basic DSP theory such as A/D converter, D/A converter, digital filtering, and frequency analysis. Our design also makes use of DSP Builder and the IP (Intellectual Property) cores provided originally for Altera's Stratix II and migrate them onto Cyclone II. This approach saves us from writing VHDL/Verilog codes for DSP functions from scratch and speeds up the design considerably.

The designed experiments can be used either in an upper undergraduate level electronics or embedded systems course or in a stand-alone DSP course for both EE and CmpE. This approach offers excellent integration of the DSP component with other components of the curriculum and achieves necessary coverage in the limited number of hours. Some parts of the designed lab materials will be used in a real-time DSP course and an embedded systems course in spring 2007. A semester-long course using all the designed materials is also planned. As DE2 board is used in several of the CmpE courses at the University of Missouri-Rolla, this approach offers students more time on DSP education rather than taking time away teaching other unfamiliar development tools.

FPGA Background

A Field Programmable Gate Array (FPGA) is a re-programmable logic device having hundreds and thousands of transistor clusters which can be used as a rapid prototyping environment. It primarily consists of logic elements (LE) which can be interconnected in a user-defined manner. These LEs can be interfaced with external hardware through I/O blocks near the outer edges of the FPGA.

The Altera DE2 board contains Altera Cyclone II 2C35 FPGA device and many supporting hardware. These include Input/Output (I/O) ports, memory, CD-quality audio codec, VGA digital-to-analog converter (DAC) and TV (NTSC/PAL) decoder, Ethernet controller, USB/RS232 interface, and various user LED, connectors, and switches. We use Quartus II v5.1 (as we have the full license for this version and is available in several computers in computer engineering labs) along with FIR mega-core plug-in v3.3 to program the DE2 FPGA boards.

Design flow of DE2 FPGA board is illustrated in Fig.1 with emphasis in designing an audio filter. We start with a design entry which is a Verilog, VHDL or Block Description File (BDF) which contains the basic pin set up and interconnections between different entities in the design. Then the audio codec configuration is integrated into the design to allow the FPGA to control the characteristics of the codec. Then the IP core, i.e. FIR compiler mega core plug-in is added into the design. Then the design is synthesized to check for programming errors. Eventually, pin assignments are made according to the pin table of the DE2 board. The Quartus project is then compiled to perform place & route, fitting and timing analysis. Then it is simulated and debugging is done with the help of Signal Tap II logic analyzer after programming the device.

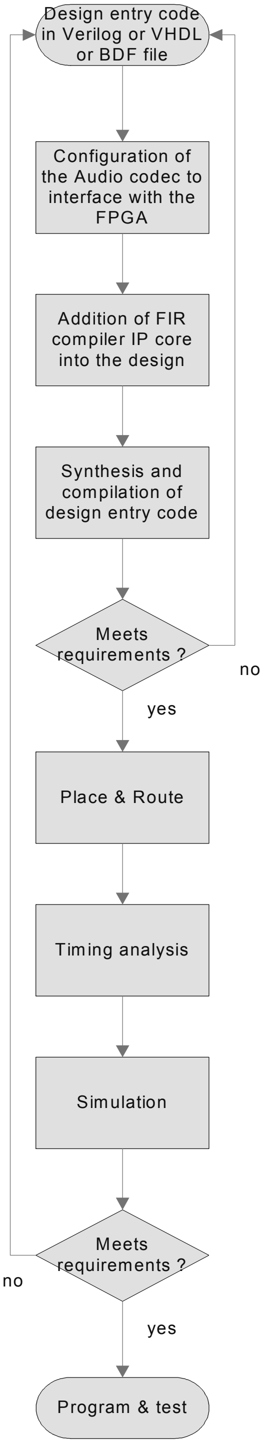

Design entry code in Verilog or VHDL or BDF file
Configuration of the Audio codec to interface with the FPGA
Addition of FIR compiler IP core into the design
Synthesis and compilation of design entry code
Meets requirements ?
no
yes
Place & Route
Timing analysis
Simulation
Meets requirements ?
no
yes
Program & test

Fig.1. Design Flow in Quartus for filter design using IP core

Laboratory Materials for DSP

We designed four DSP labs based on Altera's DE2 FPGA board. Each lab is designed to cover some aspect of DSP theory as well as FPGA operation. The summary of these four labs are described here. We also give detailed description for Lab 3 in the next section. Complete descriptions and files are available on CDROMs. Interested readers may contact the authors.

Lab 1. Audio signal generation and Run-through on DE2 board

This lab is designed to input an audio signal from the host PC's sound card or a MP3 player to DE2 board via "line-in" connector and play it back to loudspeakers. The input signal is sampled and encoded by the on-board audio codec at several different sampling frequencies and then sent directly to the decoder of the audio codec. It is output via "line-out" to loud speakers to allow students to listen to the input and output audio signals and make comparison. Aliasing is demonstrated by sampling a high frequency tone (5000 Hz) at 8 kHz sampling frequency and comparing it with a low frequency tone (1000 Hz) also sampled at 8 kHz. This lab covers the DSP theory on ADC, DAC, and sampling theorem. It also helps the students to get acquired with (or reinforce the knowledge on) the DE2 FPGA board and its related software tools.

Lab 2. Digital Signal Spectral Analysis

This lab is designed to implement Discrete Fourier Transform (DFT) or Fast Fourier Transform (FFT) on the DE2 board. Periodic rectangular or triangular signals are input to the audio codec and are transformed to frequency representation. The magnitude and phase representation is plotted to demonstrate the spectra contents of the signal. Wideband music or speech signals are also processed to illustrate their spectra. This lab covers periodic signals, random signals, Fourier transform, DFT, and spectral analysis.

Lab 3. FIR Filter

This lab is to utilize the multiplier blocks and logic arrays on Cyclone II to implement FIR filters and process audio signals before outputting them to the loudspeakers. A signal consisting of sine waves of three different frequencies is filtered by a low-pass, band-pass, or high-pass FIR filter. Thus the output signal contains only one of the three tones and can be distinguished from the input signal. Other wideband signals are also used to demonstrate the effect of filter band with and cutoff frequency concept. This lab reinforces the spectrum concept of digital signals covered in Lab 2 and demonstrates digital filter (FIR) and digital signal processing.

Lab 4. Real-Time Processing Challenges

This lab is designed to demonstrate real-time requirements for signal processing and design and implementation tradeoffs between size and speed. Lab 2 or Lab 3 is repeated with different settings in filter design as well as FPGA implementation. Higher processing speed may be achieved by using parallel architectures in the implementation at the cost of using more resources, such as memory, logic elements, and multiplier blocks, etc. With limited resources on-board, we demonstrate that there is a limit on the higher speed.

These labs can be used together as a semester-long course as an upper level undergraduate elective for both CmpE and EE curriculum. They can also be used individually as supplement to related CmpE courses such as discrete linear systems, embedded systems, and computer architecture.

Design Example- FIR Filter

Lab 3 deals with the design of a 24 bit low pass FIR filter (whose canonical form structure is described in [2]and [3]) operating at 8 kHz and which has a sampling rate for the audio codec at

8 kHz. The cut off frequency of the filter is kept at 800 Hz. Audio signal is filtered in the FPGA through an interface with audio codec on the FPGA board which accepts unfiltered audio signal through Line In and sends out filtered audio signal through Line Out.

A detailed description of critical stages of the design is explained in the following sections.

Configuring the Audio codec

The basic block diagram for interfacing the audio codec with the FPGA for designing filtering is as shown in Fig.2. The DE2 board is provided with a CD which includes a design example for capturing sound through Line In or MIC port through the audio codec. In the provided example, the input sound signal bypasses the FPGA and output directly to Line Out. It does not configure the audio codec to activate the Digital Audio Interface (DAI). In our design, the code is modified to enable the FPGA and to route the incoming data on to the FPGA for processing. The FPGA then filters the data and sends them to the D/A converter. The processed data is available for output at Line Out of the codec. A BDF file is created consisting of two block symbols. One block symbol configures the audio codec to operate as a master generating a clock out for synchronizing with the filter. The audio codec is set to operate in I2S mode. Certain areas of the

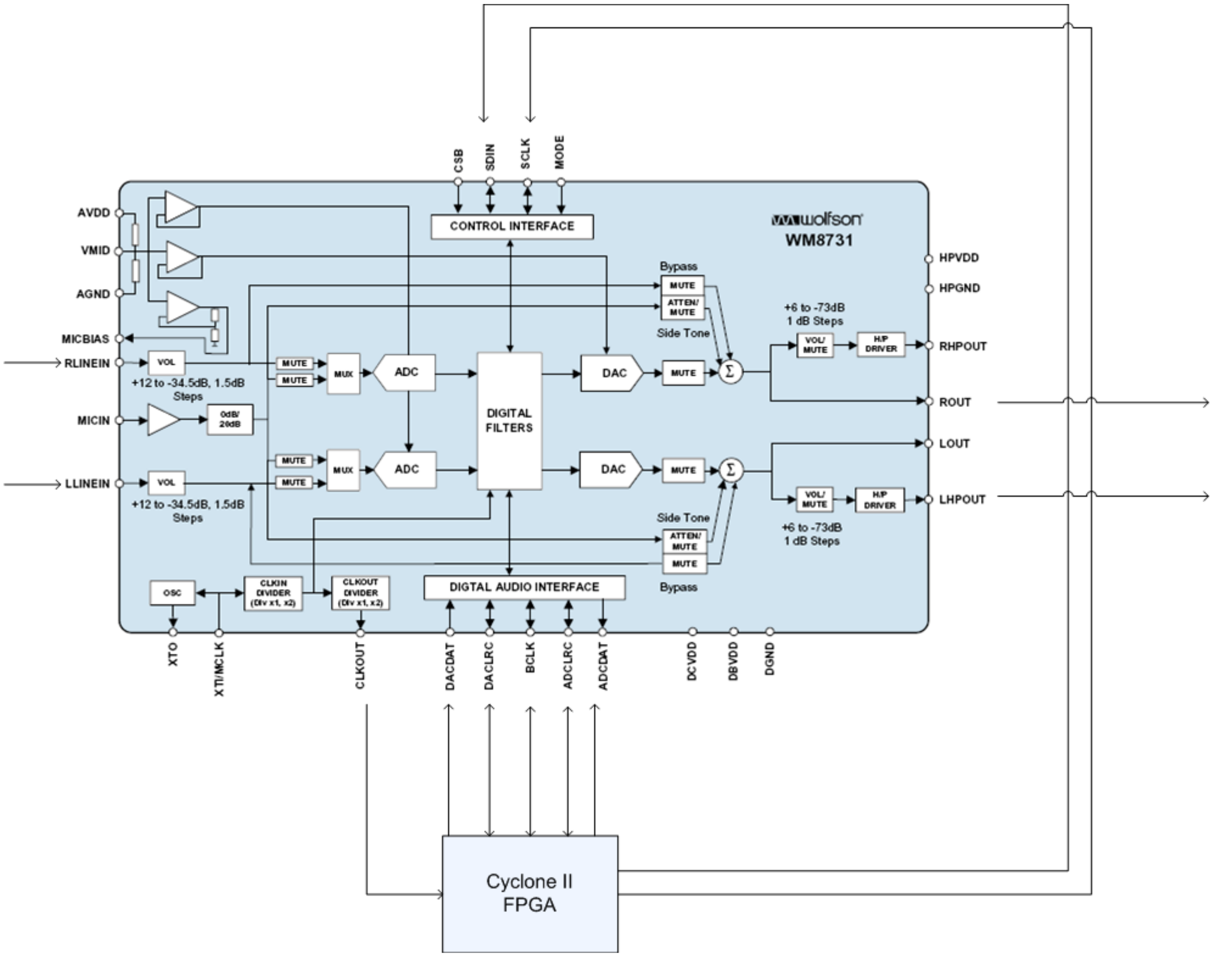


Fig.2. Interfacing the audio codec to the FPGA. Arrows indicate the flow of signals between the FPGA and the audio codec.

audio codec, such as the bypass lines, side tone circuitry, and MICIN are powered down (described in the data sheet of the audio codec). The bit length for processed data is set to its native 24 bit value. This configuration is done with the SDIN and SCLK pins of the audio codec to modify the contents of the internal register to operate in this way.

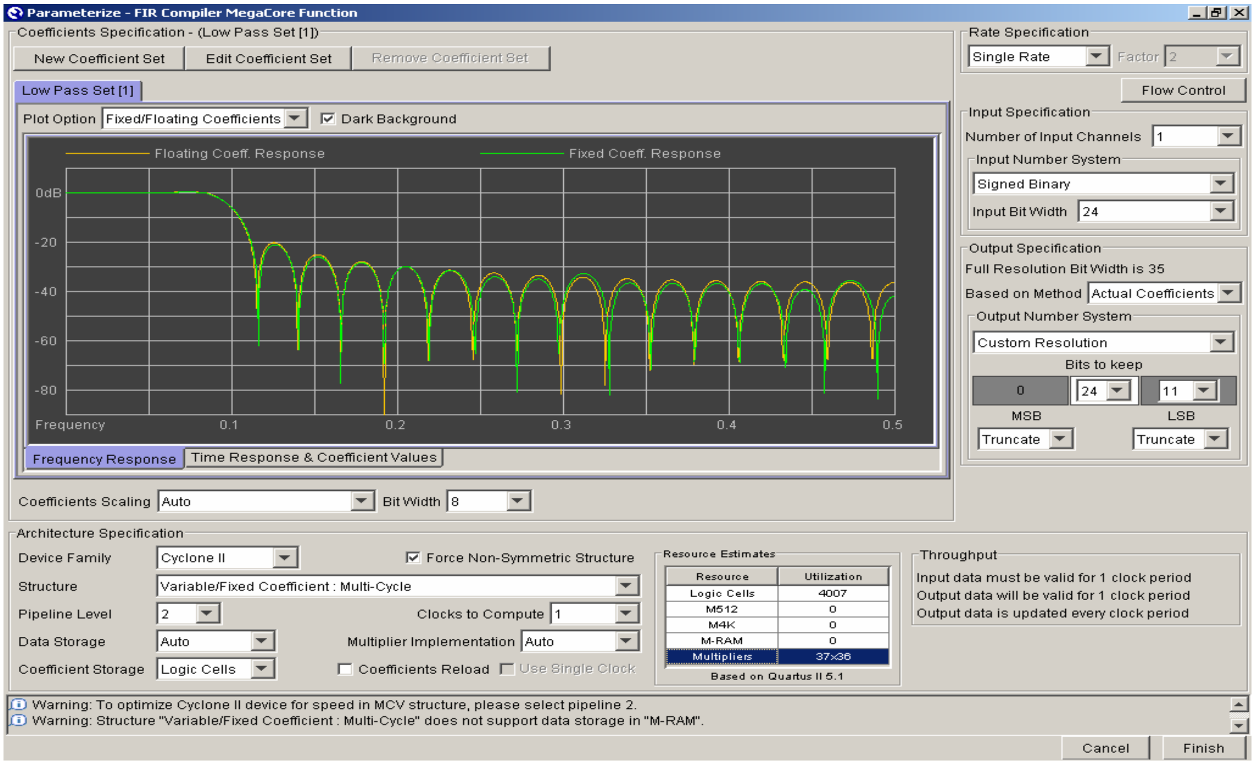
Another block symbol is generated for separating the left and right channels for the stereo input through Line In. The whole contents of this BDF file are again converted into a symbol for designing a primary BDF where the symbol for the filter from the Mega core IP plug-in can be inserted.

Adding FIR compiler IP core into the design

The top level design for the entire quartus project is generated consisting of two symbol blocks one of which contains the BDF file mentioned above and the other consists of the filter block in which the FIR compiler IP core plug-in is added.

FIR compiler mega core plug-in configuration

The FIR compiler can be invoked from the mega core plug-in wizard in Quartus and selecting the FIR compiler v3.3 core from the DSP Plug-ins and then specifying a filename for its automated code generation either in Verilog or in VHDL. The mega core plug-in offers several customization options. The structure of the filter can either be a bit-serial, fully parallel, multi-cycle or a fully serial filter. The coefficients are automatically generated with user specified bit width.



Coefficients Specification - (Low Pass Set [1])

Plot Option Fixed/Floating Coefficients ☑ Dark Background

Coefficients Scaling Auto Bit Width 8

Rate Specification

Single Rate Factor 2

Input Specification

Number of Input Channels 1

Input Number System

Signed Binary

Input Bit Width 24

Output Specification

Full Resolution Bit Width is 35

Based on Method Actual Coefficients

Output Number System

Custom Resolution

Bits to keep

0 24 11

MSB LSB

Truncate Truncate

Architecture Specification

Device Family Cyclone II ☑ Force Non-Symmetric Structure

Structure Variable/Fixed Coefficient : Multi-Cycle

Pipeline Level 2 Clocks to Compute 1

Data Storage Auto Multiplier Implementation Auto

Coefficient Storage Logic Cells ☐ Coefficients Reload ☐ Use Single Clock

Resource Estimates

Resource	Utilization
Logic Cells	4007
M512	0
M4K	0
M-RAM	0
Multipliers	37x36

Based on Quartus II 5.1

Throughput

Input data must be valid for 1 clock period
Output data will be valid for 1 clock period
Output data is updated every clock period

Warning: To optimize Cyclone II device for speed in MCV structure, please select pipeline 2.
Warning: Structure "Variable/Fixed Coefficient : Multi-Cycle" does not support data storage in "M-RAM".

Fig.3. FIR compiler Mega Core Plug-in window

The filtered data and the coefficients can either be stored on the onboard logic cells or on M4K memory that cyclone II FPGA has built into it. Since the output of the A/D converter in the audio codec is 24bit, the width for the filter is set as 24bit and the output of the filter after processing results in 35 bits. Since the resolution of the D/A converter is only 24bits, the eleven extra LSB filter output bits are truncated.

The "edit coefficient set" button offers the user access to the primary filter characteristics. Here the user can specify parameters such as the filter type, sampling rate, cutoff frequency, window type etc. This window also offers the user to either import coefficients generated by other tools such as MATLAB or a custom coefficient set can be loaded.

The sampling rate of the filter design is set at 8 KHz with a cut off frequency setting at 800 Hz. After setting up these parameters, the necessary HDL files are generated and the design is added in to the Quartus project. The resultant symbol of the FIR filter block to be used in the design is shown in Fig.4.

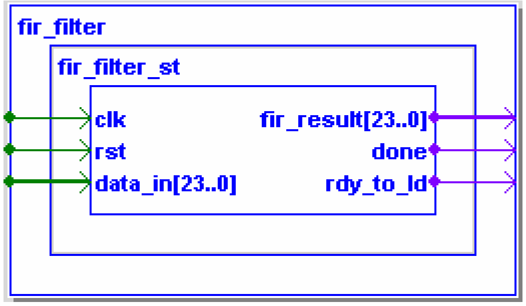
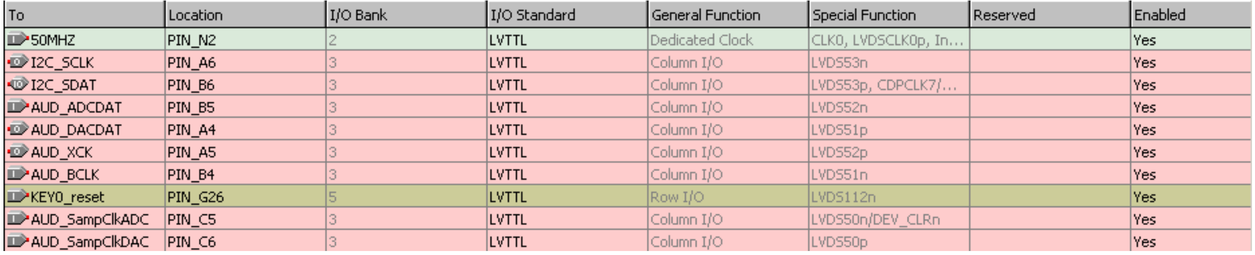


Fig.4. FIR filter block generated by Mega core plug-in Wizard manager

Compilation and Pin assignment

The pins that are placed in the BDF are now assigned to specific locations on the chip.Fig.5.



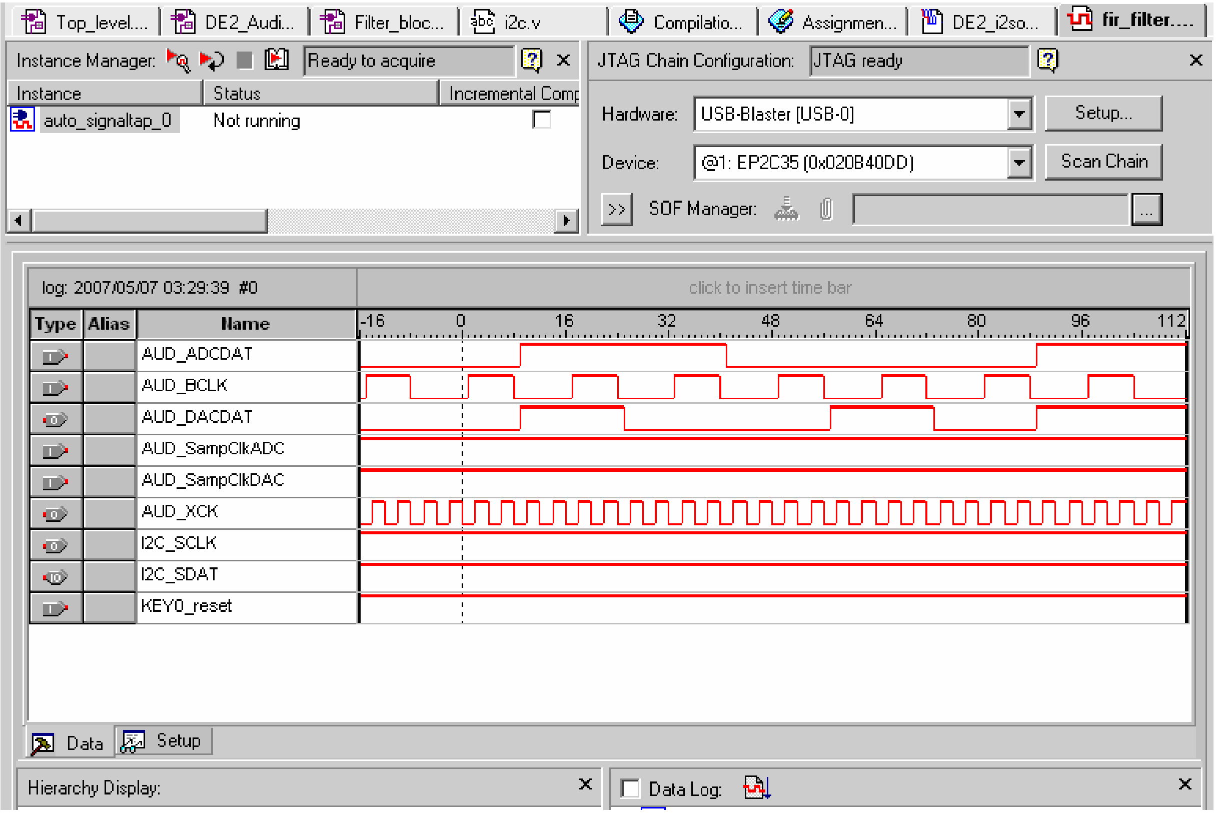
To	Location	I/O Bank	I/O Standard	General Function	Special Function	Reserved	Enabled
50MHZ	PIN_N2	2	LVTTL	Dedicated Clock	CLK0, LVDSCLK0p, In...		Yes
I2C_SCLK	PIN_A6	3	LVTTL	Column I/O	LVDS53n		Yes
I2C_SDAT	PIN_B6	3	LVTTL	Column I/O	LVDS53p, CDPCLK7/...		Yes
AUD_ADCDAT	PIN_B5	3	LVTTL	Column I/O	LVDS52n		Yes
AUD_DACDAT	PIN_A4	3	LVTTL	Column I/O	LVDS51p		Yes
AUD_XCK	PIN_A5	3	LVTTL	Column I/O	LVDS52p		Yes
AUD_BCLK	PIN_B4	3	LVTTL	Column I/O	LVDS51n		Yes
KEY0_reset	PIN_G26	5	LVTTL	Row I/O	LVDS112n		Yes
AUD_SampClkADC	PIN_C5	3	LVTTL	Column I/O	LVDS50n/DEV_CLRn		Yes
AUD_SampClkDAC	PIN_C6	3	LVTTL	Column I/O	LVDS50p		Yes

Fig.5. Pin assignments

After pin assignments the design is set for full compilation. In this stage the software maps the HDL libraries on to logic blocks, places and routes the interconnections and fits the design on to the FPGA to perform timing analysis. Any warnings resulted at this stage may be due to bad design or timing restrictions. These have to be reduced to make the design optimal. After successful compilation the software generates a chain descriptor file (CDF) which is used to program the device through USB.

Verification and Results

The proper functioning of the design after programming the FPGA can be verified in real-time with the help of Signal Tap II Logic Analyzer which is integrated into Quartus II. Thus the Signal Tap II analyzer window is to be set up before programming the FPGA. This tool allows the user to monitor the internal contents of the FPGA in real-time while it is operating. Fig.6. shows the configuration and analysis window of Signal Tap II analyzer.



Instance Manager: Ready to acquire

Instance	Status	Incremental Comp
auto_signaltap_0	Not running	□

JTAG Chain Configuration: JTAG ready

Hardware: USB-Blaster [USB-0]

Device: @1: EP2C35 (0x020B40DD)

log: 2007/05/07 03:29:39 #0

Type	Alias	Name
		AUD_ADCDAT
		AUD_BCLK
		AUD_DACDAT
		AUD_SampClkADC
		AUD_SampClkDAC
		AUD_XCK
		I2C_SCLK
		I2C_SDAT
		KEY0_reset

Fig.6. Signal Tap II logic analyzer window

The DE2 Board is then programmed using the USB cable provided with the kit. Now the FPGA is programmed to operate as an FIR filter. Sound signals fed into Line In created through software such as MATLAB can now be filtered thorough the low pass filter designed using the steps described previously. The filter response curve is similar to the characteristic curve shown in Fig.3. The filter designed in this method uses a multi-cycle structure thereby utilizing the on board multiplier blocks. The timing analysis estimate points that the maximum operating frequency for the current design is 158.6MHz without any clock skew although we operate our filter at just 8 KHz. The speed of the filter is thus limited by the performance of the audio codec. In this way many Filters with different characteristics and performance figures can easily be designed with the help of the FIR compiler Mega core plug-in.

Conclusions

Laboratory materials and experiments are designed using the Altera's DE2 FPGA for fulfilling the DSP requirement of the CmpE curriculum. Although DSP has been conventionally taught using digital signal processors or sophisticated FPGAs, we were able to successfully implement DSP filters and processing audio signals using the low-end FPGA such as the one on DE2 board. As DE2 board is used in several of the CmpE courses at the University of Missouri-Rolla, this approach offers students more time on DSP education rather than taking time away teaching other unfamiliar development tools. The designed labs will be used and assessed in a real-time DSP course in spring 2008.

Further development of lab materials may be made in areas of parallel processing with DSP, video signal acquisition, image processing techniques, and communications applications. Besides the designs on DE2 board, we are also exploiting the possibilities of using Stratix II board for our teaching and research in the DSP area. DSP education with the help of FPGAs provides invaluable information to students apart from their ease of programming. Once the curriculum is strengthened by developing more laboratory exercises its inclusion into computer engineering will provide students with state-of-the-art engineering education. Further the DE2 board would open limitless possibilities to undergraduate students in design projects.

Acknowledgements

The authors would like to thank ALTERA Corporation and their university education program for the generous donation of the DE2 FPGA boards and the prompt technical support. The work is supported in part by AFOSF under grant FA9550-07-1-0376.

References

[1] Curriculum Guidelines for Undergraduate Degree Programs in Computer Engineering, IEEE Computer Society, 2004.

[2] Dawoud, D.S., Masupe, S., “Design and FPGA implementation of digit-serial FIR filters”, AFRICON, 2004. 7th AFRICON Conference in Africa, Volume 1, 2004, pp. 203 – 209.

[3] Valls, J., Peiro, M.M., Sansaloni, T., Boemo, E., “Design and FPGA implementation of digit-serial FIR filters”, on, Volume 2, 7-10 Sept. 1998, pp. 191 – 194.

[4] Valls, J., Peiro, M.M., Sansaloni, T., Boemo, E., “A study about FPGA-based digital filters”, Signal Processing Systems, 1998 IEEE Workshop on SIPS, 8-10 Oct. 1998, pp. 192 – 201.

[5] Aziza I. Hussein, Don M. Gruenbacher, and Noureddin M. Ibrahim, “Design and Verification Techniques Used in a Graduate Level VHDL Course”, 29th ASEE/IEEE Frontiers in Education Conference, November 10 - 13, 1999 San Juan, Puerto Rico.

[6] Uwe Meyer-Baese, *Digital Signal Processing with Field Programmable Gate Array*, Springer-Verlag, New York, 2nd edition.

[7] Lab notes from Computer engineering lab of University of Missouri-Rolla available at http://ece.umr.edu/documents/112_exp_4_10_Aug06-2.pdf.

[8] José Ignacio Mateos Albiach, “Interfacing a processor in FPGA to an audio system”, Master thesis in Electronics systems at Linköping Institute of Technology, 2006.

[9] Altera literature documents: Quartus II v 5.1 Handbook, AN 306: Implementing Multipliers in FPGA Devices (ver 3.0, Jul 2004, data sheets provided with quartus software: DE2 pin table, Wolfson microelectronics WM8731 Audio codec data sheet.

Biographical information

YAHONG ROSA ZHENG
Dr. Yahong Rosa Zheng received the B.S., M.S., and Ph.D. from the University of Electronic Science and Technology of China (Chengdu), Tsinghua University (Beijing), and Carleton University (Ottawa, Canada), respectively. Her research interests include array signal processing, real-time digital signal processing, wireless communications, and wireless sensor networks. She can be reached at zhengyr@umr.edu.

SARAT KUMAR CHITNENI
Sarat K. Chitneni is a graduate student at University of Missouri-Rolla. He received his BS degree in electronics and instrumentation engineering from Andhra University, Andhra Pradesh, India in 2006. He is currently progressing towards an MS degree in Electrical engineering. His research interests include VLSI design, communications systems design and industrial process control. He can be reached at scmt9@umr.edu.

DARYL BEETNER
Dr. Beetner is an Associate Professor of Electrical and Computer Engineering at the University of Missouri-Rolla and is Associate Chair for Computer Engineering. He received his BS degree in Electrical Engineering from Southern Illinois University at Edwardsville in 1990and an MS and DsC degree in Electrical Engineering from Washington University in St Louis in 1994 and 1997, respectively. He can be reached at daryl@umr.edu.