FPGA-based Digital Signal Processing Trainer

Rosula S. Reyes, Ph.D.[1,2] Carlos M. Oppus[1,2] Jose Claro N. Monje[1,2]
Noel S. Patron[1,2] Raphael A. Gonzales[2] Jovilyn Therese B. Fajardo[2]

[1]Department of Electronics, Computer, and Communications Engineering
Ateneo de Manila University
Katipunan Avenue, Loyola Heights, Quezon City
rsjreyes@ateneo.edu, coppus@ateneo.edu, jcmonje@ateneo.edu

[2]Blue Chip Designs, Inc.
Unit 201 Xanland Place 323 Katipunan Avenue,
Loyola Heights, Quezon City
npatron@bcdph.com, rgonzales@bcdph.com, jtfajardo@bcdph.com

Abstract

Field programmable gate arrays (FPGAs) have been used in a wide range of applications including the field of digital signal processing (DSP). This paper presents the use of an FPGA in the implementation of a DSP trainer that will serve as an educational tool to effectively teach the fundamental principles of digital signal processing. This trainer is capable of performing a 1024-point discrete Fourier transform, convolution, correlation, and finite impulse response filter, which includes a low pass, high pass, and band pass filter. This paper also describes the capability of an FPGA to internally generate different input signals like a square wave, triangle wave, and a sine wave, to accept an external signal from a microphone, an MP3 player and the like, to output the transformed signal in digital or analog form, and through the use of a VGA port, to visualize the signals in a display device making this trainer low cost.

1. Introduction

With the revolutionary growth of electronics especially digital microelectronics, digital signal processing (DSP) marked its importance. It has a wide range of applications from data communication, speech, audio, or biomedical signal processing to instrumentation and robotics [1]. Because of this, there is a growing need for training systems that will effectively teach the fundamental principles of DSP. However, existing DSP trainers are quite expensive. Thus, a low cost basic DSP trainer will be beneficial especially to students in the electronics field. Moreover, the availability of these low cost DSP trainers in colleges and universities here in the Philippines will be essential to produce competitive electronics engineers in the near future.

2. Leveraging on the Benefits of FPGAs

Ten years ago, the thought of a single-chip design fulfilling all of the needs of a certain project was unimaginable. However, at present, the use of FPGAs has been a viable option.

A field programmable gate array (FPGA) is a semiconductor device containing programmable logic blocks and interconnects [2]. These interconnects are controlled by programmable switches that connect the different logic blocks in any desired configuration to implement a specific application [3]. Applications may include duplicating the functionality of certain digital circuits. Moreover, modern FPGAs can incorporate devices with simple functions like address decoders or multipliers to more complex functions like arithmetic logic units, digital signal processing, and microprocessors [4]. In effect, the trend for FPGAs is to build more intellectual property (IP) cores to facilitate the implementation of system-on-a-chip (SoC) designs [5].

In designing a product using an FPGA, a certain development process is followed. It begins with the specification of the design and ends with programming the target device or the FPGA to be used. The design is specified using a schematic capture tool or writing a source code in a hardware description language (HDL) [6],[7]. Using an HDL is usually preferred over using a schematic capture tool especially in designing large and complex circuits. Two of the HDLs endorsed by the Institute of Electrical and Electronics Engineers (IEEE) that are virtually supported by all vendors are Verilog HDL and VHDL (Very High Speed Integrated Circuit Hardware Description Language). These HDLs describe the functionality of the FPGA. Thus, there is no need to think about the gate level implementation of the circuit. Moreover, computer-aided design (CAD) tools are available that helps the designer or engineer with the rest of the FPGA design process.

With the wide-range capabilities of FPGAs and its ease of use, it has been able to tackle nearly any type of application imaginable. This includes designing FPGAs into products, using FPGAs to test products or even using FPGA-based hardware for controlling and manufacturing products [8],[9]. Thus, designing a telecommunications trainer system using an FPGA is also possible and this will be relatively cheap compared to other existing trainers.

With only an appropriate FPGA, an analog-to-digital converter (ADC), a digital-to-analog converter (DAC), and a couple of other analog components, an inexpensive DSP trainer can be implemented. For instance, get 12 one-half watt resistors and several lines of HDL, a very cheap 4096-color VGA controller is created. Another is to use the ADC and several lines of HDL code then, an inexpensive oscilloscope-like instrument can be achieved. Again, with some HDL codes and the DAC, a function generator can also be created. External function generators and oscilloscopes are also not needed because these can be generated internally by the FPGA. In addition, internal clocks can also be generated by the FPGA using phase-locked loop (PLL) circuits where most FPGAs today contain 1 to 8 of these. Thus, only basic knowledge, coding skills, and good execution is needed to come up with an innovation.

3. Costly Digital Signal Processing Trainers

Existing digital signal processing trainers are either software-based and/or hardware-based that requires the use of desktop computers to perform different DSP experiments. Also, there is a need to learn programming languages like C or Assembly to use the mentioned trainers shifting the focus of students on learning how to use the trainer and not on learning the theories behind digital signal processing. Moreover, these trainers are quite expensive and usually, it is not a stand-alone device. Additional equipments like oscilloscopes, function generators, and power supplies are also needed adding to the overall cost of using these DSP trainers. In effect, most universities and colleges here in the Philippines cannot afford to procure one.

4. DSP Trainer System

FPGAs have been used in a wide variety of applications especially in the implementation of digital systems. Some of these systems are aimed to be used for educational purposes either to provide as a substitute for current expensive learning tools or to provide another method of helping the students grasp the theoretical concepts better. An example of such a tool is the 16-bit microprocessor core implemented using an FPGA [10]. Moreover, FPGAs are becoming an attractive alternative to general-purpose DSP processors and by developing an understanding of the different signal processing algorithms, it may be efficiently implemented within an FPGA [11]. Perhaps, it is possible to implement an FPGA device that teaches the different concepts in DSP.

4.1. Overall System

The DSP trainer system is composed of a control unit, a processing unit, and a display unit as shown in Figure 1. The control unit is a keyboard connected to the PS/2 port of the processing unit. Basically, this is where the user enters his/her input for the trainer. Moreover, the processing unit contains the FPGA plus the different I/Os connected to it. Finally, the display unit is any output device (e.g. monitor, LCD, etc.) that has a VGA port.

The modules for the DSP trainer were written in Verilog HDL and it used Altera's Quartus II project navigator software for the entire design process. Then, the design was downloaded to Altera's Cyclone II FPGA.

The main modules implemented in this DSP trainer are discrete Fourier transform, finite impulse response (FIR) filter, convolution, and correlation. Moreover, under the FIR filter, a low pass, high pass, and band pass filter was also implemented.

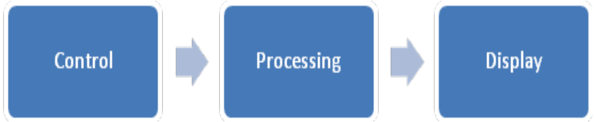

Figure 1. The DSP trainer overall system

4.2. Signal Generator

This DSP trainer does not need a function generator for its input signals because a function generator can easily be implemented in an FPGA. Thus, a module that emulates a signal generator was implemented. The output for this module serves as the input signals that can be used for the different DSP techniques. A square wave, a triangle wave, and a sine wave are internally generated.

A square wave is implemented by using a register to hold a value for a definite duration of time determined by a counter. The square wave generator inputs are a clock and a constant value called period. For every clock cycle, the counter (accumulator) increments until it is equal to half of the period. Then, the output is complemented.

A triangle wave is implemented similar to that of a square wave but instead of complementing the previous output, the value of the output is incremented or decremented by a certain factor.

The sine wave is implemented using Direct Digital Synthesis (DDS) shown in Figure 2. Initially, points of a waveform in digital format are stored in memory. These points are then recalled to generate the waveform using an address counter (accumulator) pointing to the memory. For every clock cycle, the address counter adds up depending on the desired frequency. In this trainer, the structure of the DDS is composed of a read-only-memory (ROM) containing values of one-fourth of a sine wave cycle, an address counter, and a logic circuit to invert and flip the values in the ROM to output a full sine wave cycle after running the address counter four times. By modifying the speed and the increment of the address counter, the frequency of the sine wave generated is controlled.

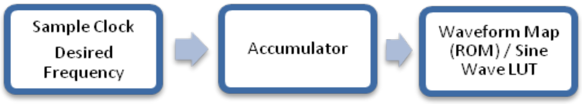

Figure 2. Block diagram of direct digital synthesizer (DDS)

4.3. Discrete Fourier Transform (DFT)

In this trainer, the discrete Fourier transform was implemented using the decimation-in-time algorithm. It is based on splitting (decimating) a particular input or sequence into smaller sequences and finding the DFT of the input from the DFTs of the decimated sequences [12].

Specifically, a 1024-point discrete Fourier transform was implemented in this trainer. This is done by first writing the 1024 data inputs in integer form is to memory. Then, each integer data saved in memory converted to single-precision floating point according to the IEEE-754 standard. The input sequence is re-ordered by dividing those values in the odd- and even-valued indices subsequently. The coefficient values needed are precomputed and stored in a look-up table (LUT). Butterfly computations then follow. The complex results of the mentioned computations are converted to its polar form giving the magnitude and phase of the DFT output.

4.4. Finite Impulse Response (FIR) Filter

This trainer implements an eight-tap low pass, high pass, and band pass FIR filter. The mentioned filters uses multiplier-adder circuits. The multiplier-adder circuit accepts pairs of input and the members of each pair are multiplied together. The product is then added to other products of all the other pairs.

The inputs to the multiplier-adder circuit are the input signal to be filtered and the filter coefficients. The filter coefficients are precomputed based on the sample rate and cutoff frequencies. These coefficients are then saved in an LUT depending on the window type chosen by the user. The available window types are rectangular, Hamming, Hanning, and Blackman.

4.5. Convolution and Correlation

In this trainer, a linear convolution and correlation are implemented. These functions almost have a similar implementation. Both uses multiplier-adder circuits. The inputs for the multiplier-adder circuits are stored in memory and the filtered signal or output is also stored back to memory. However, the difference in its implementation is the order in which the inputs are accessed from the memory.

4.6 VGA Display

This trainer is capable of displaying the input signals and the output signal of the different DSP techniques to an output device through the VGA port. There are five signals needed to display an image using the VGA port. Two signals are used to synchronize the video. These are the horizontal sync and vertical sync.

Three signals are used to control the color. These are the red signal, green signal, and blue signal [13].

Data for the red, green, and blue (RGB) signals corresponds to the pixel data. This data is saved in an external memory (e.g. 512Kbyte SRAM) connected to the FPGA. Then, data is retrieved from the RAM, formatted into lines of pixels, and sent to the display device with the appropriate horizontal and vertical synchronization pulses.

In addition, the pixel data used to form the image is accessed from the RAM using horizontal and vertical counters that are also used to generate the synchronization pulses for VGA output.

The display of the trainer is a grid-like display having two divisions. The upper grid displays the input signal while the lower grid displays the output signal of the selected modulation shown at the bottom left corner of the output device.

4.7 Using the Output Ports

The trainer has the option to output the result through the output port (GPIO 0 port) or the line-out port of the trainer. The output signal through the GPIO 0 port is in digital form of 8 bits corresponding to pins 0 to 7 with the 8th pin having the most significant bit. However, the output signal through the line-out port of the trainer is in analog form.

4.8 Using the Input Ports

The trainer is also capable of accepting external input from another trainer, from a microphone, or from other multimedia devices connected to the line-in port of the trainer. For the GPIO 1 port, the signal should be in digital form and assigned to the first 8 pins (pin0 to pin7) of the port with the 8th pin having the most significant bit. However, for both the microphone and line-in inputs, the inputs should be in analog form.

5. Summary

The FPGA-based DSP trainer was implemented using a VGA port to connect to any output device for its display and using a PS/2 port to connect to a keyboard for its control. This trainer includes the basic DSP techniques like discrete Fourier transform, low pass, high pass, and band pass FIR filter, linear convolution, and correlation.

In addition, this trainer is a stand-alone device capable of internally generating its input or acquiring it through the microphone-in or line-in ports of the trainer. Also, this trainer is capable of displaying the transformed signal in any output device with a VGA port and through the line-out port of the trainer, the transformed signal can be acquired in analog form. Overall, this FPGA-based DSP trainer has a number of capabilities as compared to other existing DSP trainer with the added benefit of being inexpensive. Thus, this trainer is an adequate resource material in teaching the basic concepts of digital signal processing.

6. Acknowledgements

This trainer was developed by Blue Chip Designs, Inc. and it was funded by the Department of Science and Technology – Philippine Council for Advanced Science and Technology Research and Development (DOST-PCASTRD).

7. References

[1] N.B. Jones, and J.D.M. Watson, *Digital Signal Processing: Principles, Devices and Applications*, IET, London, 1990.

[2] C. Maxfield, *The Design Warrior's Guide to FPGAs: Devices, Tools, and Flows*, Elsevier, New York, 2004.

[3] J.C. Whitaker, *The Electronics Handbook,* CRC Press, Florida, 1996.

[4] B. Zeidman, *Designing with FPGAs and CPLDs,* Focal Press, Massachusetts, 2002.

[5] D. Chen, J. Cong, and P. Pan, *FPGA Design Automation: A Survey,* Now Publishers Inc., Massachusetts, 2006.

[6] S. Brown and Z. Vranesic, *Fundamentals of Digital Logic with Verilog Design*, McGraw-Hill, New York, 2008.

[7] Z. Navabi, *Digital Design and Implementation with Field Programmable Devices,* Springer, New York, 2004.

[8] Al-Ayasrah, O.and A. Al-Lawama, "DSP-Based Two-Synchronized Motors Control System Using External FPGA Design", *WSEAS Transactions on Systems and Control,* Vol.3, No.4, 2008, pp. 279-288.

[9] Islam, S., N. Amin, M.S. Bhuyan, M. Zaman, B. Madon, and M. Othman, FPGA Realization of Fuzzy Temperature Controller for Industrial Application, *WSEAS Transactions on Systems and Control,* Vol.2, No.7, 2007, pp. 484-490.

[10] Alaer E., A. Tangel, and M. Yakut., "MIB-16 FPGA Based Design and Implementation of a 16-Bit Microprocessor for Educational Use", *WSEAS Transactions*

on Advances in Engineering Education, Vol.5, No.5, 2008, pp. 326-330.

[11] R.C. Cofer, and B.F. Harding, *Rapid System Prototyping with FPGAs: Accelerating the Design Process,* Elsevier, New York, 2005.

[12] M.H. Hayes, *Schaum's Outline of Theory and Problems of Digital Signal Processing,* McGraw-Hill Professional, New York, 1999.

[14] J.O. Hamblen, T.S. Hall, and Mi.D. Furman, *Rapid Prototyping of Digital Systems: Quartus® II Edition,* Springer, New York, 2006.