

VHDL Code Generator for Digital Filter Implementation

Steve Attard

Supervisor(s): Dr. Ing. Trevor Spiteri



Faculty of ICT
University of Malta

February 2018

*Submitted in partial fulfillment of the requirements for the degree of B.Sc. Computing
Science (Hons.)*

1 Introduction Chapter

1.1 Digital Signal Processing Overview

With the capabilities of modern technology and overall accessibility, it can be easily said that we are witnessing an exponential growth in multimedia and digital content consumption. It is constantly becoming easier to share and use digital data such as audio, images and video thanks to technological advances in devices such as smartphones and tablets. Although one may think of such digital media as nothing but the usual images, audio recordings and video clips, these are essentially digital signals, which can be transmitted between users to share information.

Due to this escalation in the use of digital media, the demand for digital signal processing (DSP) applications is also increasing rapidly. Digital signal processing is an area which mainly deals with the analysis and modification of digital signals. This field of study is growing in importance, as it is directly involved in anything making use of digital media, ranging from industries such as music production and filming to telecommunications, seismology and biomedicine [10] [9] [2].

As already pointed out, despite the fact that digital media can take many forms, these are basically digital signals. To fully understand what a digital signal is, the differences between analogue and digital signals must be clarified. A digital signal is a sequence of discrete values which happen in regular, equally-spaced intervals. This is a fundamental difference between digital and analogue signals, with the latter having a continuous nature. With this conceptual difference in mind, it becomes clear that these two different types of signals cannot be represented in the same domain or time-space. Analogue signals can be represented in a continuous-time domain while digital signals can be represented in a discrete-time domain. Thus, a conversion method known as sampling is needed to convert analogue signals to digital ones, so that DSP techniques can be applied on the digital signals obtained. Sampling is the process of recording the value of a continuous-time signal at specific, equally-spaced points in time. The values recorded every so often form a sequence of numbers, or equivalently, a representation of the original signal in a discrete-time domain. Once the digital signal is obtained, computers are able to analyse and modify the signal through the use of various mathematical operations.

1.2 Digital Filters

Despite the fact that digital signal processing is an area which is continuously evolving with the discovery of new technologies, some devices such as digital filters remain a widely-used, fundamental building block in many applications making use of some form of DSP [1]. Digital filters are systems which, in their simplest form, perform a technique known as convolution on an input discrete-time signal and a set of filter coefficients in order to transform the input signal and produce an output signal.

Digital filters can be categorized in two main branches, finite impulse response (FIR) and infinite impulse response (IIR) filters. Infinite impulse response filters make use of feedback, or in other words use previous output values to compute the current value of the output signal. On the other hand finite impulse response filters only use current and previous input values to determine the value of the output signal at any point in time. Thanks to this property, FIR filters can achieve linear phase and guarantee stability, making them preferred in general to IIR filters [8]. In this study, only FIR filters will be considered.

While altering a sequence of numbers which represent a signal may seem like a trivial task, there exists many different ways how to go about the filtering process and different mathematical techniques can be used to achieve the same result, each having different benefits. Furthermore, the realization of FIR filters with the desired level of accuracy is a challenging task, since the complexity of implementation increases with the number of coefficients, commonly referred to as filter order, and precision of computation [5].

The filter coefficients mentioned previously in this section are values which determine how the filter modifies the input signal, therefore the type of filter implemented is based on what values the filter coefficients have. The most commonly used types of filters include low-pass, high-pass and band-pass filters. Although this project does not revolve around filter design in particular, these types of filters will be designed using the MATLAB FDA Toolbox for implementation purposes. This tool provides an easy-to-use interface which allows the user to design a filter by specifying requirements such as filter type, filter order, design method and cut-off frequency among other options. The FDA Toolbox will be discussed at a finer level of detail in the coming sections.

Once the desired filter is designed using such a tool, a structure can be chosen to implement the digital filter characterized by the coefficients obtained. The use of Field Programmable Gate Arrays (FPGA) to implement DSP applications is gaining popularity, since these devices provide high speed thanks to their parallel architecture and offer

high reconfigurability [6]. These properties complement the requirements of digital filter implementation very well, leading to numerous efforts in producing optimized and efficient filtering mechanisms on FPGA development boards. FPGAs can be programmed using hardware description languages (HDL) such as Verilog and VHDL.

In this dissertation, a VHDL code generator for digital filter implementation is presented. Written using the Python programming language, the code generator is capable of producing generic VHDL code implementing digital filters synthesisable on FPGA. The user of the presented software can choose from four different filter structures, namely the direct-form, symmetric, transpose and distributed arithmetic structures. These structures will be explained in detail in the upcoming sections of this report. Furthermore, the code generator allows the user to specify the desired precision of computation and a VHDL test-bench is also produced, making simulation in software such as ModelSim possible. This is a convenient feature as it offers a way in which the response of the filter can be observed under certain inputs. These simulations can also be done to verify correct computation of the filtering process, by keeping a log of the simulation result and comparing it to theoretical results obtained in another Python project. Synthesis results obtained using Xilinx ISE are analysed to compare the speed and area efficiency of a number of digital filters produced by the code generator, to the efficiency of the HDL code of the same filters generated using the MATLAB FDA Toolbox.

References

- [1] DAITX, F. F., ROSA, V. S., COSTA, E., FLORES, P., AND BAMPI, S. VHDL generation of optimized FIR filters. In *Signals, Circuits and Systems, 2008. SCS 2008. 2nd International Conference on* (2008), IEEE, pp. 1–5.
- [2] GHOLAM-HOSSEINI, H., NAZERAN, H., AND REYNOLDS, K. J. ECG noise cancellation using digital filters. In *Bioelectromagnetism, 1998. Proceedings of the 2nd International Conference on* (1998), IEEE, pp. 151–152.
- [3] GOPI, M., ET AL. A VHDL implementation of direct, pipelined and distributed arithmetic FIR filters. *IJECCCE* 4, 2 (2013), 381–385.
- [4] LOSADA, R. A. Practical FIR filter design in MATLAB. *White paper The Math Works Inc* (2004).

- [5] MEHER, P. K., CHANDRASEKARAN, S., AND AMIRA, A. FPGA realization of FIR filters by efficient and flexible systolization using distributed arithmetic. *IEEE transactions on signal processing* 56, 7 (2008), 3009–3017.
- [6] NAGAKISHORE, B., KEERTHI, M., AND REDDY, K. J. FPGA implementation of distributed arithmetic for FIR filter. *International Journal of Engineering Research & Technology (IJERT)* Vol 1 (2012), 2278–0181.
- [7] PROAKIS, J. G., AND MANOLAKIS, D. G. Digital signal processing 3 rd edition, 1996.
- [8] ROSA, V. S., COSTA, E., AND BAMPI, S. A VHDL generation tool for optimized parallel FIR filters. In *Very Large Scale Integration, 2006 IFIP International Conference on* (2006), IEEE, pp. 134–139.
- [9] TEKALP, A. M. *Digital video processing*. Prentice Hall Press, 2015.
- [10] TSAO, Y.-C., AND CHOI, K. Area-efficient parallel FIR digital filter structures for symmetric convolutions based on fast FIR algorithm. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 20, 2 (2012), 366–371.
- [11] ZHOU, Y., AND SHI, P. Distributed arithmetic for FIR filter implementation on FPGA. In *Multimedia Technology (ICMT), 2011 International Conference on* (2011), IEEE, pp. 294–297.