CHAPTER 6

IMPLEMENTATION OF ADALINE IN FPGA

6.1 INTRODUCTION

The ADALINE algorithm is proposed for implementation on a Xilinx® make FPGA of type Virtex-II (Xilinx 2007). An FPGA is a semiconductor device containing programmable logic components and programmable interconnects. The programmable logic components can be programmed to implement the functionality of simple to complex combinatorial functions. In most FPGAs, these programmable logic components or logic blocks also include memory elements. A hierarchy of programmable interconnects allows the logic blocks of an FPGA to be interconnected as needed by the system designer. These logic blocks and interconnects can be programmed by the user so that the FPGA can perform the required logical function.

The signal/data flow in an FPGA takes many paths simultaneously due to the presence of an array of programmable logic components. This inherent parallelism is advantageous for implementation of ANN thereby mimicking the functioning of biological neural networks.

6.2 OVERVIEW OF AN FPGA

The typical basic architecture of an FPGA consists of an array of logic blocks and routing channels. Multiple I/O pads may fit into the height of one row or the width of one column. Generally, all the routing channels have
the same width. The interconnect wires also go to the boundary of the device where I/O cells are implemented and connected to the pins of the FPGAs. FPGAs have fast dedicated lines in between neighboring logic cells. The most common type of fast dedicated lines is carry chains. Carry chains allow creating arithmetic functions like counters or adders very efficiently. The counters and adders have low logic usage and high operating speed. All newer FPGAs have internal Random Access Memory (RAM). This has tremendously increased their scope of applications.

To define the behaviour of an FPGA, the user provides a Hardware Description Language (HDL) or schematic design. Common HDLs are Very-high-speed-integrated-circuits Hardware Description Language (VHDL) and Verilog. Then, using proprietary ‘place-and-route’ software, all the subsystems are mapped, placed and interconnected. The designer validates the map, place-and-route results via timing analysis, simulation, and other verification methodologies. Once the design and validation process is complete, a binary file is generated, using the proprietary software. This binary file is used to configure the FPGA device.

6.3 OVERVIEW OF XILINX VIRTEX-II FPGA

The proposed ADALINE algorithm is implemented on a Xilinx Virtex-II FPGA (Xilinx 2007). The Virtex-II Development Kit provides an easy to use development platform for prototyping and verifying Virtex-II based designs. The block diagram of the development kit is shown in Figure 6.1.

The Virtex-II family is a FPGA platform developed for high performance, low to high-density designs utilizing Intellectual Property (IP) cores and customized modules. The Virtex-II family supports I/O standards such as Low Voltage Differential Signal (LVDS), high performance interfaces
for Peripheral Connect Interface (PCI) and Double Data Rate (DDR) devices. The Virtex-II development kit utilizes the Xilinx 1M gate Virtex-II - XC2V1000 device.
Virtex-II devices are user-programmable gate arrays with various configurable elements. The Virtex-II architecture is optimized for high-density and high-performance logic designs. As shown in Figure 6.1, the programmable device is comprised of Input/Output Blocks (IOBs) and internal Configurable Logic Blocks (CLBs). The internal configurable logic includes four major elements organized in a regular array.

- CLBs provide functional elements for combinatorial and synchronous logic, including basic storage elements.
- Block Select RAM memory modules provide large 18 Kbit storage elements of dual-port RAM.
- Multiplier blocks are 18-bit x 18-bit dedicated multipliers.
- Digital Clock Manager (DCM) blocks provide self-calibrating, fully digital solutions for clock distribution, delay compensation, clock multiplication and division, coarse- and fine-grained clock phase shifting.

A new generation of programmable routing resources called Active Interconnect Technology interconnects all of these elements.

6.4 MODELSIM 5.8c

The development of ADALINE algorithm is simulated using the Electronic Design Automation (EDA) tool from Mentor Graphics called ModelSim ver.5.8c. Mentor Graphics was the first to combine Single Kernel Simulator (SKS) technology with a unified debug environment for Verilog, VHDL, and SystemC. The combination of industry-leading, native SKS performance with the best integrated debug and analysis environment make ModelSim the simulator of choice for both Application Specific Integrated Circuit (ASIC) and FPGA design.
The ModelSim ‘vopt’ usage mode achieves industry-leading performance and capacity through very aggressive, global compile and simulation optimization algorithms of Verilog and VHDL, improving Verilog and mixed VHDL / Verilog Register Transfer Logic (RTL) simulation performance by up to 10X. The performance mode can also improve Verilog gate-level performance by up to 4X and capacity by over 2X. ModelSim also supports very fast time-to-next simulation and effective library management while maintaining high performance with its new black box use model, known as ‘bbox’. With bbox, non-changing elements can be compiled and optimized once and reused when running a modified version of the test bench. bbox delivers dramatic throughput improvements of up to 3X when running a large suite of test cases.

6.5 EXPERIMENTAL SETUP

The block diagram of the experimental setup is shown in Figure 6.2. The experimental setup consists of the following subsystems:

- A current sensor of type CS 60-010 of Coilcraft.
- A non-inverting amplifier circuit
- An ADC circuit
- The Xilinx Virtex-II Reference board
- A PC is used as a nonlinear load as well as to develop, execute and implement the proposed ADALINE on the FPGA using ModelSim.
6.5.1 Current Sensor and the Signal Conditioning Circuit

The current sensor of type CS60-010 of Coilcraft make, explained in section 5.4.1 and an op-amp based non-inverting amplifier circuit explained in section 5.4.2 are used in this experimental setup.

6.5.2 ADC

The Xilinx Virtex-II does not have an on-chip ADC and it needs a LVDS compatible ADC for interfacing. The ADC used in this work is a 12-bit ADC of type LTC1407-1 (Linear Technology 2004). The ADC accepts bipolar signals and hence it does not require any level shifting of the signal from the noninverting amplifier mentioned in the previous section.

The LTC 1407-1 has a sampling rate of 3MSPS with two differential inputs 1.5MSPS sampled simultaneously. The device converts –1.25V to 1.25V bipolar input differentially. The absolute voltage swing for CH0+, CH0−, CH1+ and CH1− extends from ground to the supply voltage. The
serial interface sends out the two conversion results in 32 clocks for compatibility with standard serial interfaces. The block diagram of the ADC is shown in Figure 6.3.

![Figure 6.3 Block diagram of the ADC LTC 1407-1](image)

6.6 IMPLEMENTATION OF ADALINE IN FPGA

The VHDL code generated in ModelSim are real values, but implementation in FPGA does not permit real values. Hence all values are converted into integers and then implemented in Virtex-II FPGA reference board.

A training vector of $80 \times 100$ samples is formed. The number of rows in the training vector is chosen as 80 so as to compute up to 40th order harmonics. The number of columns in the training vector is chosen as 100 to accommodate 100 samples taken over one cycle of 50Hz. The training vector is computed \textit{a priori} and stored in the memory.
The load current waveform of a PC, which is a non-linear load, is converted through the ADC. The digital data from the ADC is read by the FPGA. A ZCD circuit is used to initiate the acquisition of samples. 100 samples at an interval of 200μs, spanning over one complete cycle of 20ms, are obtained. The implementation code is written in VHDL.

6.7 EXPERIMENTAL RESULTS

For the implementation of the ADALINE, code equivalent to MATLAB source code, is written using VHDL and it is implemented in Virtex-II FPGA. From the samples acquired by FPGA the load current waveform and supply voltage are reconstructed using a spreadsheet program. These are shown in Figure 6.4. It shows that the voltage is sinusoidal whereas the current wave form is not sinusoidal. The typical ModelSim execution environment on the PC is shown in Figure 6.5.

![Figure 6.4 Voltage and Current waveforms of a PC acquired by FPGA](image-url)
It is observed that the algorithm converges in 10 epochs. The convergence of error values for the first ten harmonics, A1, A2,…, A10 (out of A1 – A25) is shown in Figure 6.6.

**Figure 6.6** Convergence of error values
The convergence of the updated weight values \((w_1, w_2, \ldots, w_{25})\) of ADALINE network towards their final values is shown in Figure 6.7. For clarity, the graph shows only the first 10 values namely, \(w_1, w_2, \ldots, w_{10}\).

\[\begin{array}{cccccccccc}
& & & & & & & & & \\
& & & & & & & & & \\
W1 & W2 & W3 & W4 & W5 & W6 & W7 & W8 & W9 & W10 \\
\end{array}\]

**Figure 6.7  Convergence of weight values**

The harmonic contents in the load current waveform are computed using Equation (3.38) and given in Table 6.1. The same load current waveform is analysed by the PQA and the measured values of harmonics are listed in Table 6.1. The table also lists the values obtained from the PQA model C.A 8332 of Chauvin-Arnoux.

The harmonic values are computed up to the order of 40 as per the requirement explained in section 1.2.3. Since the data samples are handled in integer format, very small values in the vicinity of zero are obtained as zeros. Thus it is observed that the computed results for harmonics of order higher than 25 are obtained as zero values and hence the table lists the results up to the 25\(^{th}\) order of harmonics only.
Table 6.1  Comparison of results between the ADALINE in FPGA and FFT

<table>
<thead>
<tr>
<th>Harmonics Order $A_n$</th>
<th>Values of $A_n$ computed by ADALINE on FPGA</th>
<th>Readings of $A_n$ by FFT on PQA</th>
<th>Error %</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>67.96</td>
<td>68.4</td>
<td>-0.64</td>
</tr>
<tr>
<td>2</td>
<td>18.50</td>
<td>18.1</td>
<td>2.21</td>
</tr>
<tr>
<td>3</td>
<td>57.17</td>
<td>57.8</td>
<td>-1.09</td>
</tr>
<tr>
<td>4</td>
<td>11.51</td>
<td>11.6</td>
<td>-0.78</td>
</tr>
<tr>
<td>5</td>
<td>39.29</td>
<td>39.7</td>
<td>-1.03</td>
</tr>
<tr>
<td>6</td>
<td>5.04</td>
<td>4.7</td>
<td>7.23</td>
</tr>
<tr>
<td>7</td>
<td>19.70</td>
<td>19.5</td>
<td>1.03</td>
</tr>
<tr>
<td>8</td>
<td>2.73</td>
<td>2.6</td>
<td>5.00</td>
</tr>
<tr>
<td>9</td>
<td>4.29</td>
<td>5.4</td>
<td>-20.56</td>
</tr>
<tr>
<td>10</td>
<td>4.40</td>
<td>4.6</td>
<td>-4.35</td>
</tr>
<tr>
<td>11</td>
<td>5.16</td>
<td>4.2</td>
<td>22.86</td>
</tr>
<tr>
<td>12</td>
<td>4.16</td>
<td>4.1</td>
<td>1.46</td>
</tr>
<tr>
<td>13</td>
<td>5.69</td>
<td>4.3</td>
<td>32.33</td>
</tr>
<tr>
<td>14</td>
<td>2.7</td>
<td>2.85</td>
<td>-5.26</td>
</tr>
<tr>
<td>15</td>
<td>1.5</td>
<td>1.55</td>
<td>-3.23</td>
</tr>
<tr>
<td>16</td>
<td>4.0</td>
<td>5.11</td>
<td>-21.72</td>
</tr>
<tr>
<td>17</td>
<td>2.0</td>
<td>2.59</td>
<td>-22.78</td>
</tr>
<tr>
<td>18</td>
<td>3.1</td>
<td>4.25</td>
<td>-27.06</td>
</tr>
<tr>
<td>19</td>
<td>2.3</td>
<td>3.13</td>
<td>-26.52</td>
</tr>
<tr>
<td>20</td>
<td>1.5</td>
<td>2.24</td>
<td>-33.04</td>
</tr>
<tr>
<td>21</td>
<td>1.5</td>
<td>1.94</td>
<td>-22.68</td>
</tr>
<tr>
<td>22</td>
<td>1.5</td>
<td>2.77</td>
<td>-45.85</td>
</tr>
<tr>
<td>23</td>
<td>0.5</td>
<td>1.19</td>
<td>-57.98</td>
</tr>
<tr>
<td>24</td>
<td>2.99</td>
<td>1.5</td>
<td>99.33</td>
</tr>
</tbody>
</table>
The results of ADALINE on FPGA and PQA are compared in Figure 6.8. The figure shows that the ADALINE implemented on a FPGA using VHDL produces comparable results to that of FFT.

![Comparison of ADALINE on FPGA and FFT results](image)

**Figure 6.8** Comparison of ADALINE and FFT results

### 6.8 SUMMARY

The ADALINE algorithm verified using the MATLAB as explained in chapter 4 is converted to programming language C and then to VHDL. Using Xilinx ISE, the VHDL code is converted to a bit file for downloading in a FPGA. The Xilinx FPGA type Virtex-II is used to execute the proposed algorithm and implemented in the FPGA. The load current waveform is fed to the signal conditioning unit and 100 data samples are acquired from the ADC. The data acquisition time in the proposed method is 20ms.

It is observed that the time needed by the ADALINE algorithm for computing the harmonics using FPGA is 80 μs. It has been reported that the
FFT implemented on FPGA requires 102.4μs. Thus the measurement time using the proposed ADALINE implementation on FPGA reduces to 20.08ms compared to 200.1024ms using FFT algorithm implemented on a FPGA. It is also observed that the error is very small for the lower order harmonics i.e. up to 7th order. Though the error is large for higher order harmonics, their absolute values are small and hence their effects are also negligible.