FPGA Implementation of Fringe Pattern Demodulation Using the One-Dimensional Continuous Paul Wavelet Transform

Abdulbasit Abid, Umm Al-Qura University, College of Computer and Information Systems,
Computer Engineering Department, P.O. Box: 715 Makkah, Saudi Arabia

Abstract— *The novel FPGA implementation for fringe pattern demodulation using the one-dimensional Paul wavelet transform algorithm is presented. Firstly, the Paul wavelet transform is programmed using the C programming language and compiled into VHDL using the Impulse C tool. Then this VHDL code is implemented on the Altera Cyclone IV GX EP4CGX150DF31C7 FPGA. A fringe pattern image with the size of 512×512 pixels is presented to the FPGA, which processed the image using the Paul algorithm. The FPAG requires about 100 milliseconds to process the image and produce a wrapped phase map. The Paul wavelet is more suitable to phase-demodulate fringe patterns than the frequently used Morlet wavelet transform; since the Paul has better spatial localization than the Morlet wavelet transform. For performance comparison purposes, the Paul wavelet algorithm is programmed using the C language. The C code is then compiled using the Intel compiler version 13.0. The compiled code is run on a Dell Precision state-of-the-art workstation. The time required to process the fringe pattern image is one second approximately. In order to further reduce the execution time, the Paul wavelet is reprogramed using the Intel Integrated Primitive Performance (IPP) Library Version 7.1. The execution time was reduced to 660 milliseconds approximately. This confirms that at least six folds; approximately, speed up was gained using the FPGA implementation over a state-of-the-art workstation that executes heavily optimized implementation of the Paul wavelet algorithm.*

Index Terms— **Fringe analysis, Phase demodulation, Wavelet, FPGA.**

I. INTRODUCTION

Fringe pattern demodulation using digital computers has recently seen significant interest due to its widespread application in science, medicine and industry [1, 2]. Many methods for retrieving phase information from fringe patterns have been researched in considerable depth and some have now reached a mature state. For example, using phase stepping or phase shifting (PS) algorithms [3], Fourier transform profilometry (FTP) [4]. More recently windowed Fourier transform (WFT) profilometry [5] and wavelet transform profilometry (WTP) [6 - 10], have all been the topics of active research. The first of these methods requires at least three images to extract the phase of a fringe pattern, whereas the rest of the methods require only a single fringe pattern to produce a useful output measurement.

One-dimensional dimensional wavelet transform profilometry (1D-PWTP) and two-dimensional dimensional wavelet transform profilometry (2D-WTP) methods have many potential advantages when they are used for extracting

the phase of a fringe pattern, in comparison to the alternative Fourier transform and the windowed Fourier transform methods. However, the execution times of both of the WTP methods are significantly longer than those of the FTP and PS methods. These prohibitively long execution times have to date prevented the widespread uptake of wavelet transform profilometry for many practical applications and have restricted its usage to applications that require high quality measurement output but which have no limitations upon processing time. It would therefore be extremely advantageous if the execution time of the WTP methods could be significantly reduced, thus bringing their high output quality performance to bear upon a much larger set of real-world application problems.

In an effort to reduce the execution time of the 1D-WTP algorithms, the 1D-PCWT algorithm, which is the core of the 1D-WTP technique, was first programmed using the C programming language. The author has published the C source code for this algorithm, making it available in open-source form on the Internet for the benefit of other researchers [11]. The C code was compiled using the Intel C++ compiler version 13.0. The execution time required for this code to process a typical source fringe pattern image, with a size of 512×512 pixels, was measured when running on a high-performance standard computing platform and took approximately four seconds. The C code here was executed on a Dell Precision T7600 workstation with a Dual Eight Core XEON processor, clock speed of 2.7 GHz and 4 GByte of RAM.

In an effort to further reduce the execution time of the 1D-PWTP algorithm, the 1D-PCWT algorithm was reprogrammed this time using the C programming language in conjunction with the Intel integrated primitive performance library (IPP) version 7.1. The algorithm was then run on the same Dell Precision workstation. The execution time was reduced to approximately 0.6 seconds by using this optimized software implementation. However, even this optimized and reduced execution time may be considered to be far too long for many practical applications and it would be advantageous if it can be reduced still further.

II. FPGA IMPLEMENTATION

In this paper, we propose the use of a hardware implementation using an FPGA to improve the computational performance of the 1D-PWTP wavelet transform

profilometry algorithm. In order to implement the 1D-PWTP algorithm on an FPGA, the 1D-PWTP techniques must first be programmed into VHDL form and this is not a straightforward task. In order to avoid this obstacle, the author used the ImpulseC compiler [12]. This tool imposes certain restrictions upon the C language code, for example the use of pointers is prohibited. The existing C code [11] was therefore reprogrammed according to these limitations and then compiled into VHDL.

There are two basic methodologies that are used to calculate the 1D-PCWT algorithm, namely time domain and frequency domain approaches. Of these two, the frequency domain method is considerably faster and it uses the discrete Fourier transform (DFT), or fast Fourier transform (FFT). The authors have therefore used the FFT to calculate the 1D-PWTP algorithm in this implementation.

The VHDL version of the wavelet transform profilometry program was then implemented on an Altera Cyclone IV GX EP4CGX150DF31C7 FPGA [14]. This FPGA has 149,760 logic elements, 6,635,520 memory bits and 720 9-bit embedded multiplier elements. The rich combination of logic, memory, and digital signal processing (DSP) make this FPGA suitable for intensely computational DSP applications such as the design that is addressed in this paper. An Altera Cyclone IV GX FPGA development board, shown in Figure 1, was used as the hardware platform for the wavelet transform design. This development board has its own on-Board memory, consisting of: 4-MB ($\times$16) synchronous static random access memory (SSRAM), two 32-MB ($\times$32) DDR2 SDRAM, 64-MB flash and on-board clocking circuitry: 50.000-MHz oscillator, 125.000-MHz oscillator, programmable oscillator (default: 100.000-MHz). The SSRAM memory was used due its fast speed transfer, which is highly suitable for moving image contents around for processing using the wavelet-based design. All the calculations were carried out using a 32-bit float data type in both the FPGA system and also for the Dell workstation. This is to ensure that no reduction in precision occurs when using the FPGA system in comparison with the C code that is executed upon the Dell workstation.
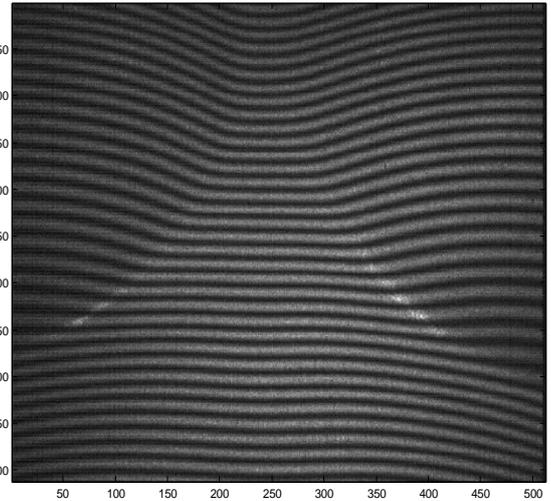

Fig. 2. A real interferometric fringe pattern image projected upon the thorax area of a female mannequin.

The FPGA was connected to a PC using Joint Test Action Group JTAG interconnectivity. The whole fringe pattern image, shown in Figure 2, is downloaded to the FPGA's external SSRAM, shown in Figure 3. The wavelet transform design was linked to a NIOS II processor with two DMA controllers to write and read both to and from the input and output buffers of the wavelet transform core and the external SSRAM. The overall design is shown in the diagram in Figure 3. The NIOS processor triggers an image DMA transfer (32-bit float 512 $\times$ 512 pixels) from the external SSRAM to the internal 1MByte input buffer, which is then read and processed by the wavelet core. Next the result is written into the 1MByte output buffer, which as a result triggers another DMA transfer from the output buffer to the SSRAM. Once the entire fringe pattern image is processed, the resultant wrapped phase map is transferred back to the PC. The output wrapped phase map that is produced by the FPGA for the input fringe pattern that was shown in Figure 2 is presented in Figure 4.
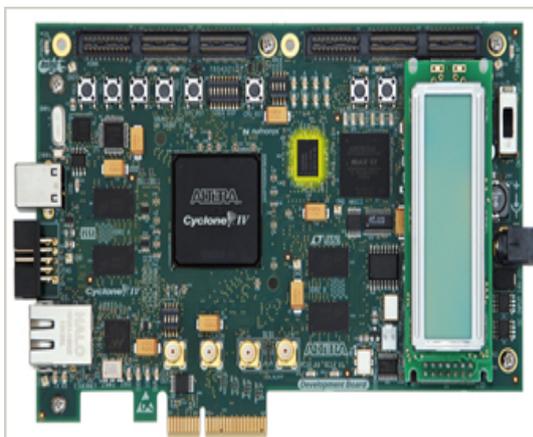

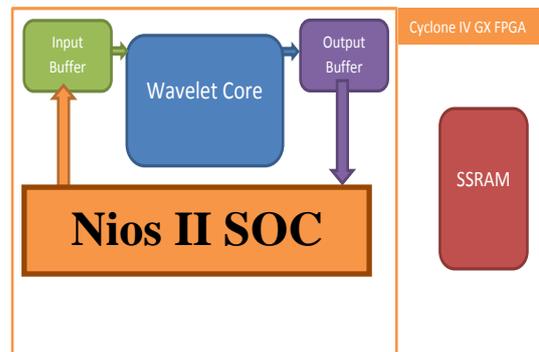**Fig. 1. Altera Cyclone IV GX FPGA development board.**

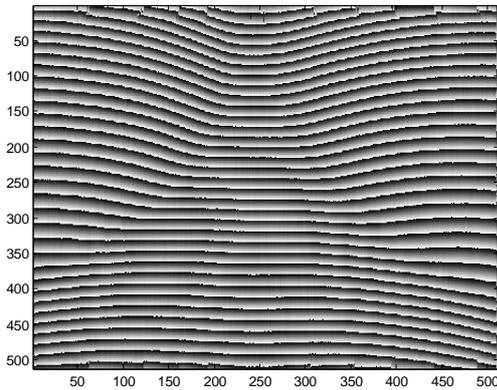**Fig. 3. FPGA system design.**

Fig. 4. FPGA generated Wrapped Phase image.

The design was compiled using the Altera Quartus II Compiler in order to achieve both an optimal design footprint on the FPGA and also fast processing speeds. The design used almost 92% of the FPGA logic elements, 55% of the FPGA on-chip memory and 70% of the FPGA 9-bit embedded multiplier elements. The FPGA design footprint on the FPGA die is shown in Figure 5. The darker blue areas represent used logic gates and the lighter blue areas represent unused logic gates of the FPGA.
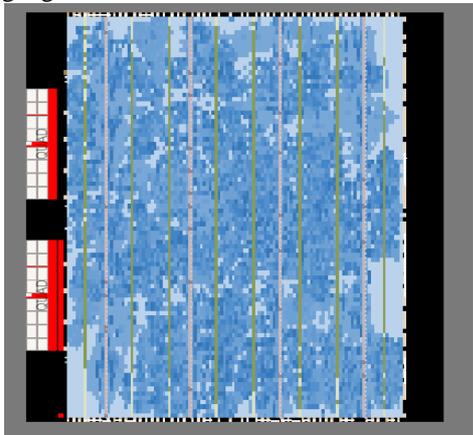

Fig. 5. The FPGA design footprint upon the FPGA die

III. CONCLUSIONS

A significant effort has been made here to reduce the execution time of the wavelet transform profilometry algorithm. Initially, the algorithm was programmed using the C language and implemented upon a state-of-the-art Dell Precision workstation using the very powerful Intel compiler. This was enhanced still further using the Intel integrated primitive's performance library (IPP). Both of these tools produce very optimized object code. The time required to execute the 1D-PWTP algorithm for a typical 512 x 512-pixel source image when using this arrangement was approximately 600 milliseconds.

In order to further reduce the execution time, the 1D-PWTP algorithm was reprogrammed using the C language and following the requirements necessitated by Impulse C. Impulse C was then used to compile this C program and produce output code in VHDL format. The VHDL code may subsequently be implemented upon a specific FPGA hardware device, namely an Altera Cyclone IV GX FPGA. Finally, a typical real 512×512-pixel fringe pattern image was processed using the FPGA approach and here the total measured execution time was reduced to approximately 100 milliseconds. This time includes the communications overhead between the PC and the FPGA. A speed up of approximately seven fold has therefore been achieved by using this hardware implementation when compared to heavily optimized conventional C code. This type of hardware acceleration makes the use of Wavelet Transform Profilometry algorithms viable for many practical applications that require higher measurement quality than is possible using established methods such as Fourier Transform Profilometry.

ACKNOWLEDGMENT

REFERENCES

[1] F. Lilley, M. Lalor, and D. Burton, "Robust fringe analysis system for human body shape measurement," Optical Engineering, vol. 39, pp. 187–195, 2000.

[2] P. Hariharan, "Applications of interferogram analysis," in Interferogram Analysis: Digital Fringe Pattern Measurement Techniques, W. R. Robinson, and G. T. Reid, eds. (Philadelphia: Institute of Physics, 1993), pp. 262–284.

[3] K. Creath, "Temporal phase measurement methods," in Interferogram Analysis: Digital Fringe Pattern Measurement Techniques, W. R. Robinson, and G. T. Reid, eds. (Philadelphia: Institute of Physics, 1993), pp. 94–140.

[4] M. Takeda, H. Ina, and S. Kobayashi, "Fourier-transform method of fringe-pattern analysis for computer-based topography and interferometry," J. Optical Society of America, vol. 72, pp. 156–160, 1982.

[5] K. Qian, "Two-dimensional windowed Fourier transform for fringe pattern analysis: Principles, applications and implementations," Optics and Lasers in Engineering, vol. 45, pp. 304-317, 2007.

[6] M. Gdeisat, M. Lalor, and D. Burton, "Spatial carrier fringe pattern demodulation by use of a two-dimensional continuous wavelet transform," Applied Optics, vol. 45, pp.8722-8732, 2000.

[7] J. Zhong, and J. Weng, "Spatial carrier-fringe pattern analysis by means of wavelet transform: wavelet transform profilometry," Applied Optics, vol. 43, pp. 4993-4998, 2004.

[8] M. Gdeisat, A. Abid, D. R. Burton, M. J. Lalor, F. Lilley, C. J. Moore, and M. Qudeisat, "Spatial and Temporal Carrier Fringe Pattern Demodulation using the One-Dimensional Continuous Wavelet Transform: Recent Progress, Challenges and Suggested Developments", Optics and Lasers in Engineering, Special Issue: Wavelet Processing, vol. 47, pp. 1348-1361, 2009.

[9] A. Z. Abid, M. A. Gdeisat, D. R. Burton, M. J. Lalor, and F. Lilley, "Spatial fringe pattern analysis using the two-dimensional continuous wavelet transform employing a cost function," Applied Optics, vol. 46, pp. 6120-6126, 2007.

[10] A. Z. Abid, M. A. Gdeisat, D. R. Burton, and M. J. Lalor, "Spatial fringe pattern analysis using the Paul wavelet transform," in proceedings of the SPIE 7000, France, 2008.

[11] M. Gdeisat, and F. Lilley, "Fringe Pattern Analysis Using Wavelet Transforms and Phase Demodulation Using Wavelet Transform," http://www.ljmu.ac.uk/GERI/79684.htm

[12] "Impulse Accelerated Technologies," www.impulseaccelerated.com.

[13] "Cyclone IV GX FPGA Development kit," http://www.altera.com/products/devkits/altera/kit-cyclone-iv-gx.html.

AUTHOR BIOGRAPHY

The Author is an assistant Professor in the Computer Engineering Department at Umm Al-Qura University, Makkah, Saudi Arabia. He has been awarded a Doctor of Philosophy in Image Processing from Liverpool John Moores University, UK in 2008. He has studied Master of Science in Computer and Information Networks at University of Essex, UK in 2004. He has also studied Master of Science in Electrical Engineering at Colorado State University, USA in 2003. He has gained Bachelor in Electrical and Computer Engineering from Umm Al-Qura University, Makkah, KSA. His research interests are fringe pattern analysis, wavelet transform and FPGA.