Hindawi Publishing Corporation
X-Ray Optics and Instrumentation
Volume 2008, Article ID 168237, 7 pages
doi:10.1155/2008/168237

Research Article

Custom Hardware Processor to Compute a Figure of Merit for the Fit of X-Ray Diffraction Peaks

Juan A. Gomez-Pulido,[1] Florentino Sanchez-Bajo,[2] Sidolina Pereira dos Santos,[3] Miguel A. Vega-Rodriguez,[1] and Juan M. Sanchez-Perez[1]

[1] *Department of Technologies of Computers and Communications, Polytechnic Institute, University of Extremadura, Campus Universitario s/n, 10071 Caceres, Spain*
[2] *Department of Applied Physics, School of Industrial Engineering, University of Extremadura, Avenida de Elvas s/n, 06071 Badajoz, Spain*
[3] *Department of Computer Science, Polytechnic Institute of Leiria, Alto do Vieiro, 2401-951 Leiria, Portugal*

Correspondence should be addressed to Juan A. Gomez-Pulido, jangomez@unex.es

Received 24 October 2007; Revised 5 February 2008; Accepted 24 March 2008

Recommended by Scott Misture

A custom processor based on reconfigurable hardware technology is proposed in order to compute the figure of merit used to measure the quality of the fit of X-ray diffraction peaks. As the experimental X-ray profiles can present many peaks severely overlapped, it is necessary to select the best model among a large set of reasonably good solutions. Determining the best solution is computationally intensive, because this is a hard combinatorial optimization problem. The proposed processors, working in parallel, increase the performance relative to a software implementation.

1. Introduction

Peaks in X-ray diffraction patterns can be modelled using analytical functions. In this way, the widely used pseudo-Voigt function [1] represents, in many cases, a good approximation to the real shape of the X-ray diffraction profiles. This is especially important when strong overlap between adjacent reflections arises, and it is necessary to separate the contribution of the different peaks, for example, to obtain information about microstructural quantities (as the mean size domain) from the parameters describing the shape and width of the peak [2–4]. For severely overlapped reflections, many peak combinations (with different parameters) can exist generating the same or similar experimental profile. For this reason, a figure of merit quantitatively assesses the fit to the experimental data. This is a typical combinatorial optimization problem. Due to the fact that there are not any exact methodologies to find the optimal solution, evolutionary algorithms (EAs) are used for this purpose. Nevertheless, the repetitive, multistep computation of the figure of merit required by the algorithm makes it necessary to search for technologies that accelerate the overall process. In this paper, a reconfigurable hardware custom processor is proposed. Although this processor alone is not competitive, the ease with which the hardware can be configured as distributed processors working in parallel increases the final performance even compared to software implementation on cluster computers.

2. The Optimization Problem

In diffraction profile, several peaks can appear, perhaps many of them severely overlapped. After obtaining the experimental data (usually in the form of intensity versus angle), and determining how many peaks are in a particular region of the spectrum, the goal is to fit a model profile to the data. In this regard, if a profile has two overlapped peaks, many two-peak sets generate model profiles that fit reasonably well to this experimental profile. In order to determine the best solution, a function whose value evaluated for each solution allows quantifying the degree of fit to the experimental profile is needed. This function is called the figure of merit.

2.1. Generation of the Tiffraction Profile

In order to test the procedure depicted here, we have used as benchmark the simulated step-scanning diffraction profile shown in Figure 1. This profile has been generated between 25° and 35° with a step of 0.01°, assuming CuKα radiation, and including two strong overlapped peaks modelled by means of the pseudo-Voigt function. In order to perform a realistic simulation, we included the following:

(1) the CuKα_2 component of each reflection, generated as a pseudo-Voigt function with the same shape and width as the CuKα_1 component, a maximum peak intensity equal to the maximum peak intensity of the CuKα_1 divided by 2, and a position of the maximum equal to $x_{02} = 2\cdot\arcsin[\sin(x_{01}/2)\cdot\lambda_2/\lambda_1]$, where $\lambda_1 = 1.5405929$ Å and $\lambda_2 = 1.5444274$ Å are, respectively, the wavelengths of the CuKα_1 and CuKα_2 radiation, and x_{01} the position of the maximum of the CuKα_1 peak;

(2) a linear function representing the background, in the form $b(x) = 100 - 10\,[(x/25) - 1]$;

(3) statistical noise (assuming a Poisson distribution).

The pseudo-Voigt function is defined as

$$\mathrm{pV}(x) = I_0[\eta\, L(x) + (1-\eta)\, G(x)], \tag{1}$$

where $L(x)$ and $G(x)$ are, respectively, the Cauchy and Gauss functions;

$$L(x) = \frac{1}{1 + ((x - x_0)/w)^2},$$
$$G(x) = e^{-((x-x_0)/w)^2 \cdot \ln 2}. \tag{2}$$

Then, the pseudo-Voigt function is

$$\mathrm{pV}(x) = I_0\left[\frac{\eta}{1 + ((x - x_0)/w)^2} + (1-\eta)\, e^{-\ln 2\cdot((x-x_0)/w)^2}\right], \tag{3}$$

where I_0 is the maximum intensity of the peak, x_0 is the position of the maximum, w is of half width at half maximum (HWHM), and η is the partition factor, a number between 0 and 1 that indicates the relative weight of the Cauchy and Gauss functions (see Figure 2).

The parameters characterizing the two pseudo-Voigt functions describing the CuKα_1 peaks included in the profile were the following: I_0 (1) = 1000.0, η (1) = 0.5, w (1) = 0.2°, x_{01} (1) = 30.0°, I_0 (2) = 500.0, η (2) = 0.5, w (2) = 0.2°, and x_{01} (2) = 30.5°. Here, the variable x corresponds to the angular variable 2θ, usually expressed in degrees.

2.2. The Figure of Merit

In the present work, the test profile with two strong overlapped peaks was considered for experimental purposes.

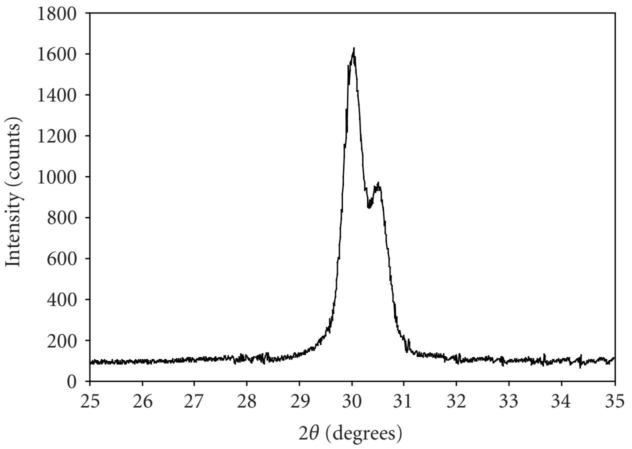


Figure 1: The simulated diffraction profile with two very strong overlapped peaks used in this work.

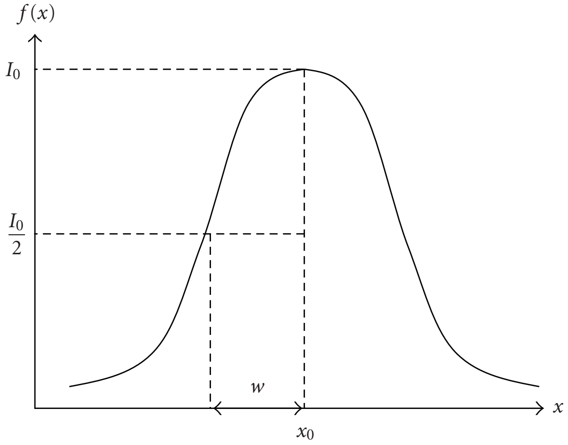


Figure 2: Meaning of some parameters defining the pseudo-Voigt function.

The objective of the optimization process is to establish how many peaks there are (although in our case this information is assumed in the generation of the profile) and what are the parameters characterizing the pseudo-Voigt functions (I_0, x_0, η, and w) that provide the best fit to the data. In other words, it must be determined which one of the many two-peak sets (possible solutions) represents the best fit to the experimental profile. For this purpose, a figure of merit to calculate the degree of adjustment to the profile is needed. With this aim, one could define several figures of merit. According to the usual practice in X-ray diffractometry, in this work, we have chosen the chi-squared function, χ^2, given by the following expression:

$$\chi^2 = \sum_{i=1}^{N} p_i\left(y_{i,\mathrm{obs}} - y_{i,\mathrm{cal}}(\vec{\alpha})\right)^2, \tag{4}$$

where the following hold.

(1) N is the number of data points, in our case, 1001.

(2) $y_{i,\mathrm{obs}}$ is the value of the measured intensity at the ith step. In our case, this intensity corresponds to the simulated profile.

(3) $y_{i,\mathrm{cal}}$ is the value of the intensity calculated from the model at the ith step. The calculated intensities are obtained from the model using two pseudo-Voigt functions to represent the CuKα_1 peaks (and the corresponding CuKα_2 components for each peak) and a linear background.

(4) $p_i = 1/\sigma_i^2 = 1/y_{i,\mathrm{obs}}$ is the weight assigned to each intensity, assuming Poisson statistic.

(5) $\vec{\alpha} = (\alpha_1, \alpha_2, \alpha_3, \ldots)$ is a vector representing the parameters of the problem to be optimized. In this case, the parameters are I_0, x_0, w, and η for each peak, in addition to the coefficients of a linear function describing the background.

The lower the figure of merit is, the closer the solution is to the objective. Therefore, the effort of the problem consists of finding the vector $\vec{\alpha} = \{I_0(1), x_0(1), w(1), \eta(1), I_0(2), x_0(2), w(2), \eta(2), \ldots\}$ so that χ^2 be minimum.

The number of peaks in the profile (and even their approximated position) is usually a priori known data, because the crystallographic structure of the material analyzed by means of X-ray diffraction is often known. But also it would be interesting to know if it is possible to determine the number of peaks using a computational tool. It is necessary to keep in mind that an optimization can be good but false if spurious peaks are added, because the decrease of the degrees of freedom can lead to a very good fit without physical meaning. Obviously, the optimization can also be bad if any peak is omitted. However, this is outside the scope of this work.

2.3. Using Evolutionary Algorithms

The presented problem has a very high number of possible solutions, and it is necessary to find the best one. There are not any exact methodologies to find the optimal solution of this problem, where conventional algorithms do not locate good results. For this reason the authors of this paper have used evolutionary algorithms (EAs) [5, 6], because these algorithms (in sequential and parallel implementations [7]) have been employed with success in many works. Nevertheless, some considerations should be followed to use evolutionary techniques for this optimization problem: the individual formulation (containing the $\vec{\alpha}$ information) must be carefully chosen and the evolutionary algorithm must avoid falling in a local minimum. This way, two EAs have been used to tackle this optimization problem: genetic algorithm and differential evolution algorithm.

Genetic Algorithms (GAs) imitate the way nature selects the best individuals [8, 9]. They frequently are used on complex optimization problems whose set of possible solutions is very large. The algorithm basically consists of an iterative three-step loop. Each loop is considered to be a generation. Each step is characterized by a genetic operator: crossover, selection, or mutation, that works (in this order) in each generation. Differential evolution (DE) is a less known evolutionary algorithm [10]. From 1994, DE has been used for many optimization problems, with satisfactory results. It is used here with the goal of comparing it with the GA results. DE is a very simple population-based stochastic function minimizer/maximizer, used in a wide range of optimization problems, including multi-objective optimization [11].

2.4. Difficulties

Computing EAs can be difficult because the task usually requires great computational effort. If a high-level of precision is required, many computing resources could be necessary, with their associated cost. For this reason, any tool that helps to accelerate the computations is welcome. One of the appropriate technologies for this purpose is reconfigurable computing.

3. Hardware Custom Processor

3.1. Reconfigurable Computing

Reconfiguration of circuitry at runtime to suit the application at hand has created a promising paradigm of computing that blurs traditional frontiers between software and hardware. This powerful computing paradigm, named reconfigurable computing, is based on the use of programmable logic devices, mainly field programmable gate arrays (FPGAs) [12] incorporated in board-level systems. FPGAs have the benefits of hardware speed and software flexibility; also, they have a price/performance ratio much more favorable than application-specific integrated circuits (ASICs). For these reasons, FPGAs are a good alternative for many real applications in image and signal processing, multimedia, robotics, telecommunications, cryptography, networking, and computation in general [13].

Furthermore, as reconfigurable computing is becoming an increasingly important computing paradigm, more and more tools are appearing in order to facilitate the FPGA programmability using higher-level of hardware description languages (HDLs).

3.2. Goal

The main goal of this research is to determine if it is profitable to develop a specific-purpose FPGA processor that computes the evolutionary algorithm in order to find the best solution of the optimization problem. Since the biggest resource consumption comes from the arithmetic computation of the figure of merit, the goal is to design an arithmetic processor to compute it. In the design of this processor, it is convenient to introduce the largest possible degree of parallelism, because this is the main advantage of the hardware to increase the efficiency in comparison with the software.

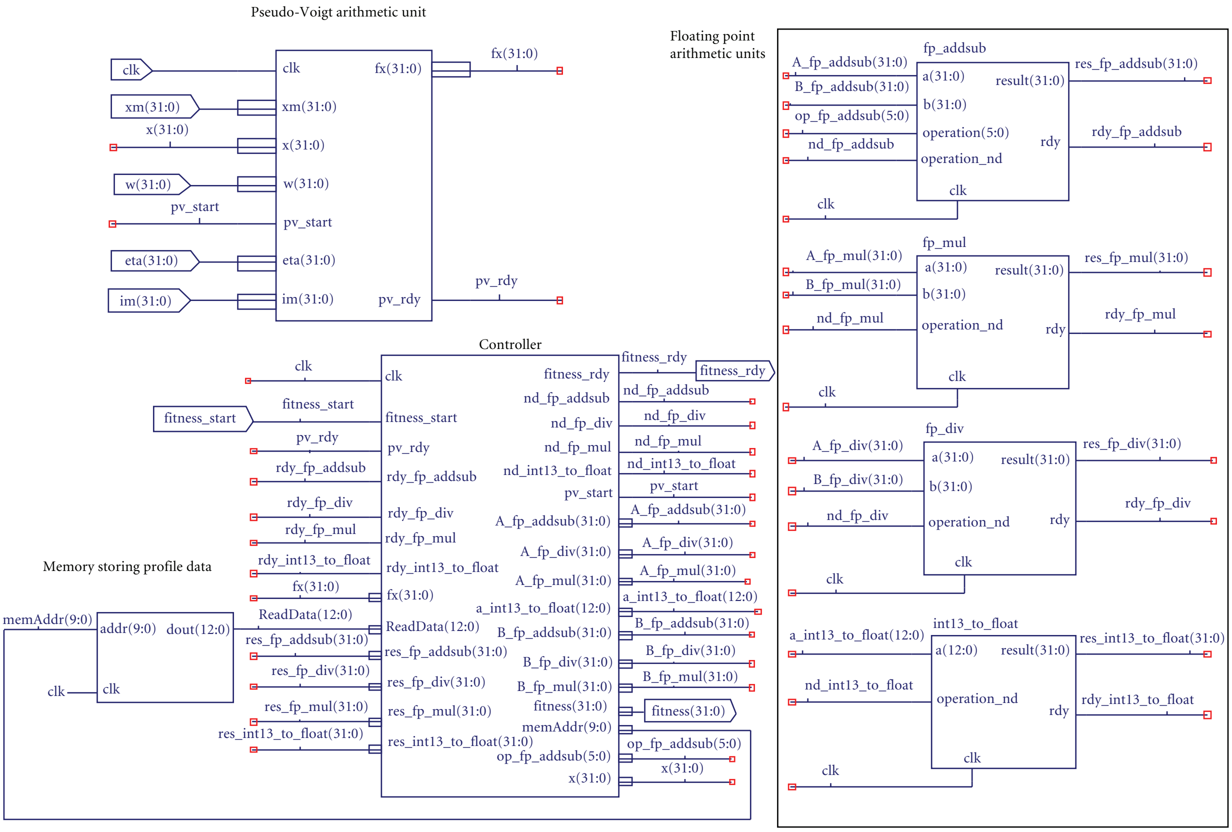


FIGURE 3: Top-level architecture of the processor.

3.3. Designed Processor

A processor computing the figure of merit has been designed. In order to probe this processor, a test circuit implementing the benchmark profile gives the input values of the parameters ($I_0\,(1) = 1000.0$, $\eta\,(1) = 0.5$, $w\,(1) = 0.2$, $x_{01}\,(1) = 30.0$, etc.), and reads the output signals (a fitness_rdy signal indicates when merit computation has been finished; and a merit signal indicates the value of the figure of merit).

The top-level architecture of the processor is shown in Figure 3. This architecture has the following parts: a memory storing the data of the profile to be processed, a set of floating-point arithmetic units, a control unit, and a pseudo-Voigt (pV) arithmetic unit.

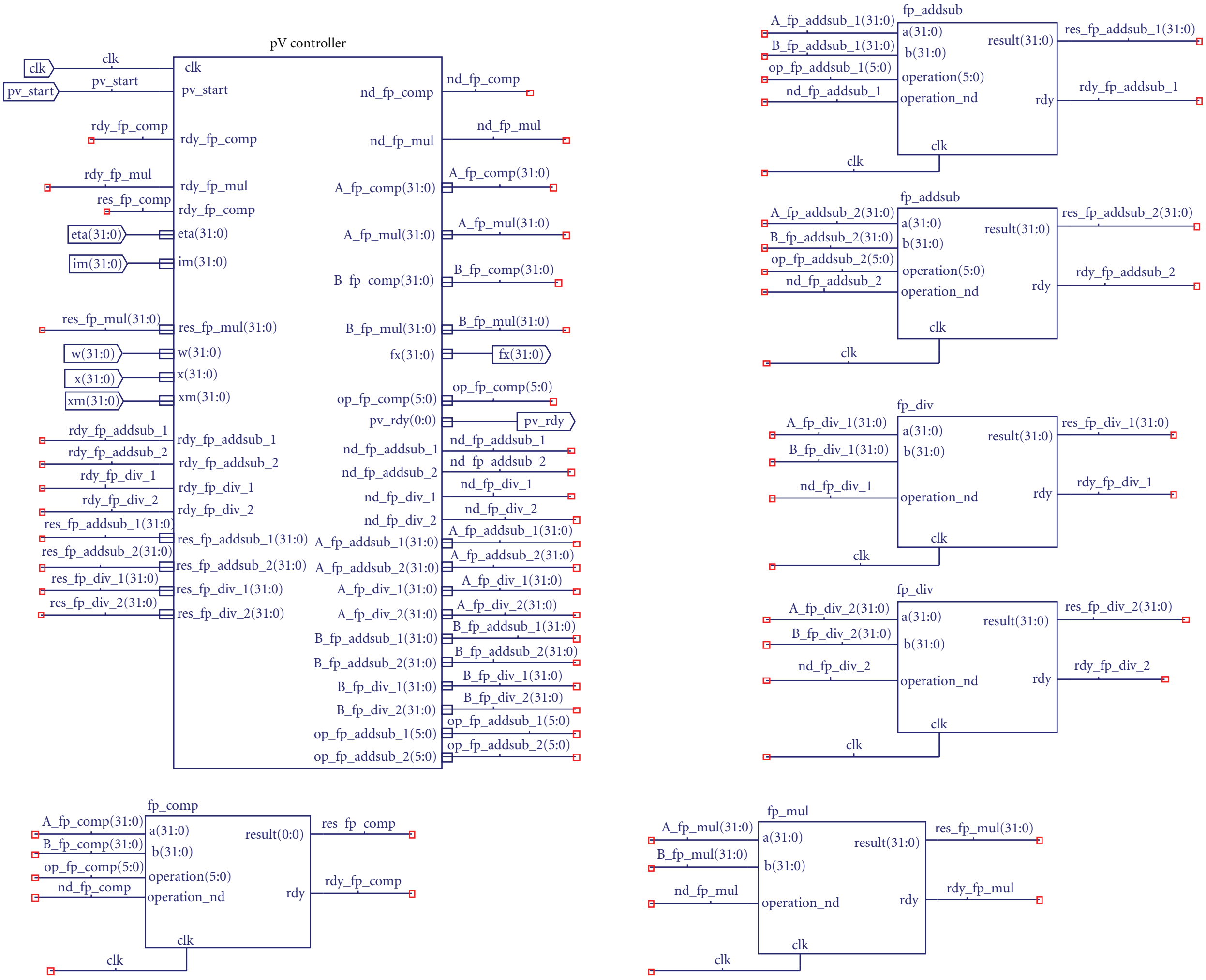
The memory storing the data of the profile to be processed is a Read-Only-Memory (ROM) device. It stores the data (counts versus degrees) of the profile used as benchmark (Figure 1). The size of this memory is 832 bytes (512 addresses for 13-bit words). Any profile can be loaded into this memory through an initialization file.

The set of floating-point arithmetic units perform the operations described in 4. This set is formed by four units: one adder/subtracter, one multiplier, one divider, and one converter of integer numbers to floating-point format. The real numbers processed in these units have a precision of 32 bits.

The controller is the core element of the processor. It plays the role of controlling the sequence of calculation of the figure of merit. The controller needs to read the profile data coming from the memory, and give them, in the appropriate order, to the floating-point arithmetic units, setting accurate control signals. This unit is also connected with the pV arithmetic unit. Finally, the calculated value of the figure of merit is read from this control unit.

A design direction that would contribute to an increase in performance is introducing a high grade of parallelism. Theoretically, this is possible in the processor because from (4) the circuit could carry out N adders in parallel. But, due to the FPGA limited size, to introduce N floating adders, N memories and N pV arithmetic units are not possible. Therefore, the parallelism advantages of the hardware must come from this processor.

The pV arithmetic unit computes the operations described in (3) according to the input signals that arrive from the test circuit. This unit consists of two parts, as it is shown in Figure 4: a set of floating-point arithmetic units and a pV controller. The floating-point arithmetic units are necessary to run the operations described in (3). There are six units: two adder/subtracters, two dividers, one multiplier, and one converter of integer numbers to floating-point format. The real numbers processed in these units also have a precision of 32 bits. There is a reason for which several repeated units of the same floating-point arithmetic module have



(a) (b)

Figure 4: Architecture of the pseudo-Voigt arithmetic unit.

been included: the controller can run parallel operations, with an increase in the final efficiency. The duplication of some of these modules has a low-occupation cost for the FPGA. The pV controller drives the necessary operations of the pV arithmetic unit. It controls the calculus sequence of the operations described in (3) in a parallel way, thanks to the duplicated floating-point units. In Figure 5, the parallel operations are shown. The operations corresponding to the exponential part in (3) (step 5b in Figure 5) have been computed by means of an iterative algorithm, with parallel operations too. Finally, the calculated value of the figure of merit is read from the output of the pV controller by the merit controller.

3.4. Design Tools and Prototyping Platform

The processor has been designed with Xilinx ISE 9 environment [14]. The merit controller and the pV controller are described by means of the Handel-C hardware design language [15]. The floating-point arithmetic units and the memory were generated from the Xilinx IP CoreGenerator tool. In order to test the system in a real hardware platform, the Enterpoint (http://www.enterpoint.co.uk/), Broadown2 board has been used. This board ships a Xilinx Spartan 3 2000 FPGA [16]. For this device, the designed processor uses the 28% of the FPGA resources. Finally, we have checked that the processor results in the prototyping board match with the simulated results in the developed software.

3.5. Performance

In general terms, one can say that the performance of the processor is its computation time in comparison with that of a software application running on a computer. To get the maximum potential performance of the processor, it is necessary to optimize its internal architecture. For this reason, two aspects have been considered: the influence of the parallel operations in the arithmetic calculus of the

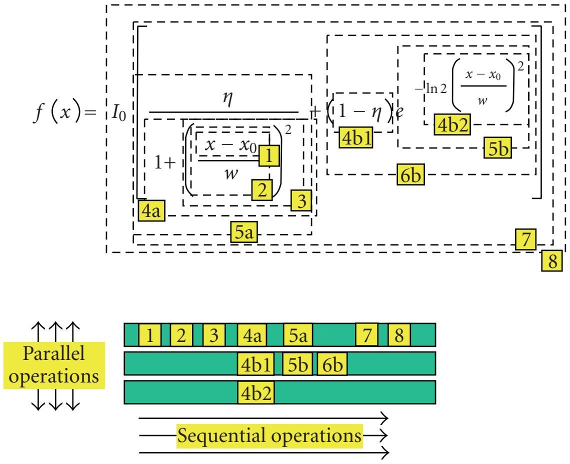


Figure 5: Parallel operations in the the pseudo-Voigt function. The calculus sequence is divided into 8 steps, numbered in yellow boxes. The steps #4 and #5 consist of 3 and 2 parallel operations, respectively.

pseudo-Voigt function and the agreement between precision and performance of the exponential function in the hardware implementation.

Increasing to the maximum the number of parallel operations in the arithmetic calculus of the pseudo-Voigt function, it has been possible to pass from 108 MHz to 112 MHz the operation frequency, that which means to pass from 11 microseconds to 9 microseconds the run time of the processor. These quantities are measured for the FPGA device above-mentioned. In conclusion, it can be said that the increase of the parallelism allows obtaining a higher-operation frequency, when decreasing the number of sequential operations. The result of the value of the figure of merit is unchanged.

The pseudo-Voigt function has an exponential term (labelled as 5b in Figure 5) whose hardware implementation has a strong influence in the overall performance. This term is computed by means of an iterative algorithm with a predefined number of iterations. For more iterations, the precision is higher, although starting from certain number of iterations the precision does not improve significantly. For example, if 30 iterations are used, the value of the figure of merit is 135 598.06 and it is reached in 19 microseconds. But if only 10 iterations are considered, its value is 135,593.52 and the time consumption is reduced to 11 microseconds. A compromise must be made between precision and performance.

If the figure of merit is evaluated by means of software running on an Intel Core2 Duo 2 GHz-based personal computer (PC), the time used is 1 microsecond. If the figure of merit is evaluated with a custom processor implemented on an FPGA Xilinx XC3S2000-5-fg456 device running at 110 MHz, the used time is 9 microseconds. Although a single processor is not competitive, the ease of using reconfigurable hardware as distributed processors working in parallel increases the final performance vastly. This can be observed in Figure 6, where a comparison of performances is shown.

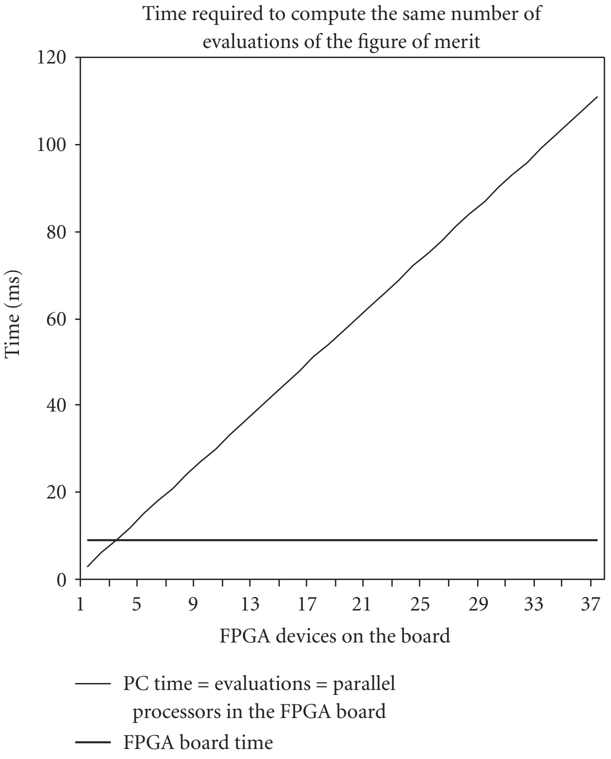


Figure 6: Time comparison between an FPGA board (with a number of FPGA devices implementing parallel processing of the figure of merit) and one PC.

The processor occupies the 28% of the considered FPGA. Therefore, a lot of parallel processors cannot be used in the same FPGA in order to multiply the performance. The hardware solution based on FPGA would be interesting in case massive computations are required in order to release the host computer to dedicate itself to other tasks. But the best strategy consists in using several FPGA devices in the same board implementing many processors that can make parallel computations for the optimization algorithm. In this case, the advantage on the software is assured. It should be kept in mind that the evolutionary algorithms work with populations, which consist of individuals whose values of the figure of merit should be evaluated in parallel. For this reason, the use of parallel processors adapts perfectly to the features of the optimization problem, besides increasing the performance notably. But a cost study is necessary in order to see the viability of the reconfigurable solution.

For example, as we can see in Figure 6, the performance of the reconfigurable solution is equal to the obtained one in the considered PC when three FPGAs (implementing 9 parallel processors) are used. Assuming that the price of the design and fabrication of a simple custom board is around $200, and that the price of only one XC3S2000 FPGA device is $58, the cost of the hardware solution ($374) is almost equal to the compared computer ($350) for the same result. Increasing the number of FPGAs starting from here, the performance increases notably (Figure 6) and the cost reduces more and more in comparison with a cluster solution

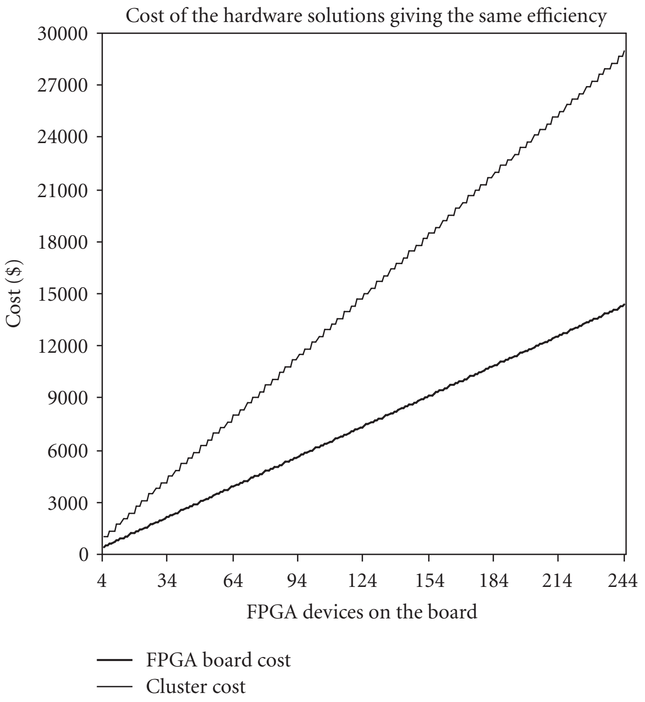


Figure 7: Cost comparison between the FPGA board and a cluster with as many nodes as necessary to reach the same efficiency as the FPGA board.

offering the same performance, as we can see in Figure 7. Nowadays, a code running on a multiprocessor cluster is an alternative solution, but with a much bigger cost. Supposing we want the evaluation of 246 figures of merit, we can build a custom board with 82 FPGAs obtaining 9 microseconds and with a price of $4,956. In the opposite case, a multiprocessor cluster with 28 of those PCs as nodes working in parallel and evaluating, 9 figures of merit each can give the same time, but it has a price around $10 100.

4. Conclusions and Future Works

We can conclude, with the obtained results shown in Figures 6 and 7, that the more complex the massive computation is required, the more effective the solution based on reconfigurable hardware is together with its cost advantage. The complexity of the considered diffraction profiles determines the computational complexity.

On the other hand, since minimizing the figure of merit is the first goal of this research, it would be interesting in the future to introduce multiobjective optimization [17]. This is a technique that permits to reach several optimization objectives at the same time. For example, multiobjective optimization can minimize the figure of merit, search the optimal number of peaks if the number of overlapped peaks in a determined region of the profile is not known, and maintain the problem parameters within predefined ranges. Also, another important step in our next research is to know if it is possible to determine the number of peaks using a computational tool.

Acknowledgment

This work has been supported in part by the Spanish Government under grant TIN2005-08818-C04-03 (project OPLINK).

References

[1] P. Thompson, D. E. Cox, and J. B. Hastings, "Rietveld refinement of Debye-Scherrer synchrotron X-ray data from Al_2O_3," *Journal of Applied Crystallography*, vol. 20, part 2, pp. 79–83, 1987.

[2] Th. de Keijser, E. J. Mittemeijer, and H. C. F. Rozendaal, "The determination of crystallite-size and lattice-strain parameters in conjunction with the profile-refinement method for the determination of crystal structures," *Journal of Applied Crystallography*, vol. 16, part 3, pp. 309–316, 1983.

[3] S. Enzo, G. Fagherazzi, A. Benedetti, and S. Polizzi, "A profile-fitting procedure for analysis of broadened X-ray diffraction peaks. I. Methodology," *Journal of Applied Crystallography*, vol. 21, part 5, pp. 536–542, 1988.

[4] F. Sánchez-Bajo and F. L. Cumbrera, "The use of the Pseudo-Voigt function in the variance method of X-ray line-broadening analysis," *Journal of Applied Crystallography*, vol. 30, part 4, pp. 427–430, 1997.

[5] D. B. Fogel, *Evolutionary Computation. Toward a New Philosophy of Machine Intelligence*, IEEE Press, Piscataway, NJ, USA, 1995.

[6] Th. Bäck and H.-P. Schwefel, "An overview of evolutionary algorithms for parameter optimization," *Evolutionary Computation*, vol. 1, no. 1, pp. 1–23, 1993.

[7] E. Alba and M. Tomassini, "Parallelism and evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 5, pp. 443–462, 2002.

[8] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Boston, Mass, USA, 1989.

[9] J. R. Koza, *Genetic Programming: On the Programming of Computers by means of Natural Evolution*, MIT Press, Cambridge, Mass, USA, 1992.

[10] K. Price and R. Storn, "Differential evolution: numerical optimization made easy," *Doctor Dobb's Journal*, vol. 22, no. 4, pp. 18–24, 1997.

[11] H. A. Abbasse and R. Sarker, "The pareto differential evolution algorithm," *International Journal on Artificial Intelligence Tools*, vol. 11, no. 4, pp. 531–552, 2002.

[12] B. Zeidman, *Designing with FPGAs and CPLDs*, CMP Books, Gilroy, Calif, USA, 2002.

[13] M. A. Vega-Rodriguez, J. M. Sánchez-Perez, and J. A. Gómez-Pulido, "Guest editors' introduction—special issue on FPGAs: applications and designs," *Microprocessors and Microsystems*, vol. 28, no. 5-6, pp. 193–196, 1994.

[14] Xilinx ISE 9.2i software manuals, http://www.xilinx.com/.

[15] K. Ramamritham, K. Arya, and G. Fohler, "System software for embedded applications," in *Proceedings of the 17th International Conference on VLSI Design (VLSID '04)*, pp. 12–14, IEEE Computer Society, Mumbai, India, January 2004.

[16] Spartan-3 FPGA family: complete data sheet, http://www.xilinx.com/.

[17] Y. Sawaragi, H. Nakayama, and T. Tanino, *Theory of Multiobjective Optimization*, Academic Press, Orlando, Fla, USA, 1985.