UNIVERSITY SAAD DAHLAB OF SCIENCE AND TECHNOLOGY OF BLIDA

Faculty of Engineer sciences

Electronics Department

A memory submitted in conformity with the requirements for the fulfillment of degree of

MAGISTERE

In Electronics (Option: Image & Speech)

FPGA IMPLEMENTATION OF 2D SIGNALS ENCODER/DECODER USING QMF BASED DYADIC DWT FAST WAVELET MALLAT'S ALGORITHM: APPLICATION TO RADIOISOTOPES IMAGES AND NEUTRON TOMOGRAPHY PROJECTIONS

By:

SAADI Slami

Jury:

Mr. H.SALHI	M.C, USTSD,BLIDA	President
Mr. A. GUESSOUM	Professor, USTSD, BLIDA	Supervisor
Mr.M.OULED ZMIRLI	M.C, CU, MEDEA	Examiner
Mr.M.MAMOUNE	C.C, USTSD,BLIDA	Examiner
Ms.N.BENBLIDIA	C.C, USTSD,BLIDA	Invited
Mr. A.GOUGAM	C.C, UMB, Boumerdès	Invited
Mr. M.TOUIZA	A.R, CRNB, COMENA	Co-Supervisor

Blida, November 2007

To my parents

To my brothers and sisters

To my wife and my children

To all my friends

Abstract

In this work, we present an Implementation of 2D signals Encoder/Decoder using dyadic Discrete Wavelet Transform based on quadrature mirror filters (QMF), by applying fast wavelet Mallat's algorithm. The proposed work is implemented using Matlab software, compiled on an embedded development kit (EDK) using standard C language on Xilinx Virtex-II FPGA board.

The calculated wavelet coefficients will be encoded using Huffman coding scheme in order to be transmitted progressively through an Ethernet TCP/IP based connection. The proposed design can significantly accelerate the DWT and the possible reconfiguration can be exploited to reach a higher performance in the future.

The basic motivation of our application to radio isotopic images and neutron tomography projections is to reduce the data volume and to achieve a low bit rate in the digital representation images without perceived loss of image quality.

The system is designed to be integrated as an extension to the nuclear imaging system implemented around our nuclear research reactor and can be used to accelerate other image processing applications.

Résumé

Dans ce travail, nous présentons une implémentation d'un codeur/décodeur pour les signaux 2D en employant la transformé des ondelettes discrète dyadique basé sur les filtres à miroir quadrature (QMF), en appliquant l'algorithme rapide de Mallat. Le travail proposé est mis en application en utilisant le logiciel Matlab, compilé sur un kit de développement embarqué (EDK) en utilisant le language de programmation C standard et implementé sur la carte FPGA Virtex-II de Xilinx .

Les coefficients d'ondelettes calculés seront codés en utilisant le code de Huffman afin de les transmettre progressivement dans une connection réseau ethernet TCP/IP. La conception proposée peut accélérer d'une manière significative la DWT et la possibilité de reconfiguration peut être exploitée pour atteindre une meilleure performance dans le futur.

Notre application à la tomographie neutronique et à l'imagerie par radio isotopes est motivée par la réduction de la taille des images (projections 2D) prises dans un milieu hostile (due à la radioactivité) et la possibilité de les transmettre avec un bas débit binaire (low bit rate) sans nuire à leur qualité.

Ce travail est conçu pour être intégré dans le système d'imagerie nucléaire mis en oeuvre autour de notre réacteur de recherche, et peut être employé dans d'autres applications de traitement d'image.

ملخص

في هدا العمل, نقترح تصميم و انجاز مشـفر و مفكك للشفرة من أجل الاشارات دات بعدين و هدا باستعمال المحول الرقمي الثنائي للمويجات المكون من مرشحات دات تناظر رباعي QMF. بتطبيق خوارزمية مالات Mallat. العمل المقترح تم تصميمه باستعمال برنامج ماتلاب و تمت ترجمته الى الالكترونيك الضمني باستعمال لغة برمجة العتاد C و انجازه على بطاقة Virtex-II FPGA الصادرة عن شركة Xilinx.

يتم تشفير معاملات المويجات المحسوبة باستعمال مشفر هافمان Huffman من أجل ارسالها تدريجيا عبر شبكة اتصال تعمل بالبروتوكول TCP/IP . يمكن للتصميم المقترح تسريع عملية التحويل الرقمي للمويجات مع امكانية استغلال اعادة تشكيل التصميم من أجل الحصول على نجاعة أفضل في المستقـبل.

تطبيق هدا العمل في مجال الطوموغرافيا النوترونية و في التصوير بالنظائر المشعة جاء بدافع تقليص حجم الصور النووية والاسقاطات النوترونية المأخودة في وسط دي خطورة بسبب ارتفاع درجة الاشعاعات المؤينـة. يمكن أيضا زيادة سرعة الارسال بتخفيض التدفق الرقمي للمعطيات (الصور) بدون التأثير على جودة الصور الملتقطة.يدخل تصميم و انجاز هدا العمل في اطار نظام التصوير النووي المراد تركيبه حول مفاعل البحث (السلام)، و يمكن استعماله في تطبيقات أخرى لمعالجة الصور.

ACKNOWLEDGEMENTS

I would like to express my sincere thanks to my supervisor, Professor A. Guessoum for his friendly welcome to me and my friends to the LATSI laboratory and for his open and warm attitude, which made me, feel like being at home during my studies at Blida University. Also, I am indebted to my mentors, in Electronics Department for their attention, time and flexibility in classroom lectures, to understand the area of signal/image processing, especially to the department head Mr.B.Kazed and Post-Graduation responsible Mr.K.Kara.

Honest thanks to Mr. H.Salhi for accepting the presidency of the jury, sincere gratitude also to Mr.M.Ouled Zmirli, Mr.M.Mamoune and Ms.N.Benblidia who honoured me by examining this work, thanks also to Mr.A.Gougam for accepting our invitation to the jury.

Particular gratitude to my co-supervisor and friend M. Touiza for his constant encouragement and support in hard times, as well as all good friends for their efforts, during all our research works realized in the Nuclear Research Center of Birine, specially: B.Mohammedi, DJ.Khelfi, A.Bouzidi, M.Salhi, A.Messai, and H.Mekki.

I am grateful to C.Belaragueb, RASIC group responsible, and I. Abdellani, Director of Nuclear Instrumentation Division, for accepting me as a member in this group.

Thanks to Mr. A. Kerris, General Director and Mr. A. Benazza, General Secretary, for their kindness and helpful nature. Special thanks go to Mr. S.AIT-MOHAMMED, Reactor Director, for his free and friendly manner.

I would also like to thank all of my colleagues in the university and in the Nuclear Research Center (CRNB) for their friendship, encouragement and technical advice, and who have made my time enjoyable by providing a pleasant environment.

As it is impossible to mention in this brief acknowledgment every single person, I would like to thank all those who have contributed in a way or another to this achievement.

I owe my deepest gratitude to my parents for their sacrifices, their support and their love; they gave me everything in life. I can not find the words to thank my wife for her devotion, endurance, patience, understanding and love. She stood besides me and inspired me during my life. This achievement is also hers.

LIST OF FIGURES AND TABLES

1. LISTE OF FIGURES :

2. LIST OF TABLES :

CONTENTS

CHAPTER 1

GENERAL INTRODUCTION

Digital image compression is a very popular research topic in the field of multimedia processing. The major focus of research is to develop different compression schemes/algorithms in order to provide good visual quality and fewer bits to represent an image in digital format. These compression schemes are either implemented in software using high level languages or in hardware using application specific integrated circuit (ASIC).

Traditionally signal and image processing applications are implemented either in general-purpose Programmable Digital Signal Processors (DSPs) or built using Application Specific Integrated Circuit (ASIC) technology. Typically, Programmable DSP processors contain built-in Multiply/Accumulate units (MACs) and use a high level instruction set to implement signal processing algorithms ranging from simple designs such as Finite Impulse Response (FIR) and Infinite Impulse Response (IIR) filters to more complex designs including audio, image and video processing **[1]**.

Recently reconfigurable devices namely field-programmable gate arrays (FPGA) have been introduced to implement DWT compression schemes/algorithms in hardware as a flexible and cost effective solution. Although current FPGA implementations based on discrete cosine transform (DCT) support video frame formats. The performances of these devices have improved and it is common nowadays to see FPGA chips replacing their DSP counterparts. These advancements in density and performances provided some exciting options for DSP designers. FPGA technology offers an attractive combination of low cost, high performance combined with an apparent flexibility, while maintaining the advantages of custom functionality like an ASIC. Furthermore, FPGAs can be reconfigured repeatedly and their configurations upgraded to match any changes in the design, the ASIC technology lacks this latter property.

The wavelet transform has emerged as an alternative to the popular Fourier Transform (FT) and its related transforms, particularly the Discrete Cosine Transform (DCT).The first factor of the wavelet transform popularity is its localization in time and frequency. Secondly, if there is only a DCT and a FT transforms, there exist a multitude of wavelet transforms. These wavelets belong to different families and

present different characteristics; this variety gives the designer the flexibility to choose the wavelet that suits a particular application. The third factor is its ability to unveil any hidden details within a signal by looking into it at different resolutions. This latter property is better known as multi-resolution analysis. Generally, wavelet transforms are implemented using a filter banks. It consists of a low pass filter and a high pass filter combined with either a decimation by a factor of 2 (analysis) or an interpolation by a factor of 2 (synthesis).

For tomography, the gray level value represents the relative linear attenuation coefficient of the object. The general framework for radiological image compression is similar to other digital compression fields; the framework includes three major stages: image transformation, quantization, and entropy encoding.

Due to the data volume and to achieve a low bit rate in the digital representation without perceived loss of image quality, the demand for transmission bandwidth and storage space in the digital radiology environment increases.

The basic motivation of our application to radio isotopic images and neutron tomography projections is to reduce the data volume and to achieve a low bit rate in the digital representation images without apparent loss of image quality, in order to reach good 3D reconstruction on a distant computer or storage for future retrieval.

The system is designed to be integrated as an extension to the nuclear imaging system implemented around our nuclear research reactor and can be used to accelerate other image processing applications. The design had to be implemented in reconfigurable hardware. Furthermore, the performance of the hardware module had to be analyzed and compared against the software implementation version.

This thesis is organized into six chapters. The aim is to present information in a concise and comprehensive manner so that the reader can understand necessary theoretical concepts and the practical work.

In **Chapter 2** an overview of the wavelet transform is provided. First, the overview highlights the historical developments related to the transform alongside with the basic mathematical theory behind it. To show the effectiveness of the wavelet transform in the field of signal and image processing some popular algorithms used for its computation are also described. Finally, wavelet transform-based compression scheme is presented.

Chapter 3 introduces the concept of configurable computing and programmable hardware. After a very brief overview of programmable hardware, the chapter

focuses on different FPGA architectures with a special emphasis on Xilinx devices and more particularly on the VIRTEX II family (the FPGA used in this work).

In **Chapter 4** we give a brief description of Es salem Research Reactor and its utilisation for many applications, such as neutron tomography and radio isotopes production for nuclear imaging.

Chapter 5 Describes the design and Implementation of the Encoder/Decoder system. An encoding/Decoding system is implemented on MATLAB and some results are reported with showing performances.

In **chapter 6**, A mixed software/hardware implementation of our system is carried out and presented using the EDK (Embedded Development Kit) supplied with the FPGA based board. The proposed system uses the MicroBlaze feature of the Xilinx Virtex-II in order to generate a platform system running software applications developed with standard C language on FPGA hardware. This possibility allows us to implement the image Encoder/Decoder with standard C and run it in the designed platform. Results and performance evaluation are carried out and included in this chapter with comments.

At the end, some concluding remarks are provided and possible developments of the current work are highlighted.

CHAPTER 2

THE WAVELET TRANSFORM: AN OVERVIEW

2.1- Introduction

In recent years, the wavelet transform emerged in the field of image/signal processing as an alternative to the well-known Fourier Transform (FT) and its related transforms, namely, the DCT and the Discrete Sine Transform (DST). In the Fourier theory, a signal (an image is considered as a finite 2-D signal) is expressed as a sum, theoretically infinite, of sines and cosines, making the FT suitable for infinite and periodic signal analysis. For several years, the FT dominated the field of signal processing, however, if it succeeded well in providing the frequency information contained in the analysed signal; it failed to give any information about the occurrence time. The first step in the long research journey was to cut the signal of interest in several parts and then to analyse each part separately. The idea at a first glance seemed to be very promising since it allowed the extraction of time information and the localisation of different frequency components. This approach is known as the Short-Time Fourier Transform (STFT). The fundamental question, which arises here, is how to cut the signal? The best solution to this problem was of course to find a fully scalable modulated window in which no signal cutting is needed anymore. This goal was achieved successfully by the use of the wavelet transform.

Formally, the wavelet transform is defined by many authors as a mathematical technique in which a particular signal is analysed (or synthesised) in the time domain by using different versions of a dilated (or contracted) and translated (or shifted) basis function called the wavelet prototype or the mother wavelet. However, in reality, the wavelet transform found its essence and emerged from different disciplines and was not, as stated by Mallat, totally new to mathematicians working in harmonic analysis, or to computer vision researchers studying multiscale image processing **[9]**.

At the beginning of the 20th century, Haar, a German mathematician introduced the first wavelet transform named after him (almost a century after

the introduction of the FT, by the French J.Fourier). The Haar wavelet basis function has compact support and integer coefficients. Later, the Haar basis was used in physics to study Brownian motion **[10]**. Since then, different works have been carried out either in the development of the theory related to the wavelet, or towards its application in different fields. In the field of signal processing, the great achievements reached in different studies by Mallat, Meyer and Daubechies has allowed the emergence of a wide range of wavelet-based applications. In fact, inspired by the work developed by Mallat on the relationships between the Quadrature Mirror Filters (QMF), pyramid algorithms and orthonormal wavelet bases **[9]**. However, the most important work was carried out by Ingrid Daubechies. Based on Mallat's work, Daubechies succeeded to construct a set of wavelet orthonormal basis functions, which have become the cornerstone of many applications **[11]**. Recently, JPEG2000, a biorthogonal wavelet-based compression has been adopted as the new compression standard **[12]**.

2.2- Continuous Wavelet Transform

Different ways to introduce the wavelet transform can be envisaged. However, the traditional method to achieve this goal remains the use of the Fourier theory (more precisely, STFT). The Fourier theory uses sine and cosine as basis functions to analyse a particular signal. Due to the infinite expansion of the basis functions, the FT is more appropriate for signals of the same nature, which generally are assumed to be periodic. Hence, the Fourier theory is purely a frequency domain approach, which means that a particular signal f(t) can be represented by the frequency spectrum F, as follows:

$$F(\omega) = \int_{-\infty}^{+\infty} f(t) e^{-j\omega t} dt \qquad 2.1$$

The original signal can be recovered, under certain conditions, by the inverse Fourier Transform as follows:

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} F(\omega) e^{j\omega t} d\omega \qquad 2.2$$

Obviously, discrete-time versions of both direct and inverse forms of the Fourier transform are possible.

Due to the non-locality and the time-independence of the basis functions in the Fourier analysis, as represented by the exponential factor of equation 2.1, the FT can only suit signals with ñtime-independentò statistical properties. In other words, the FT can only provide global information of a signal and fails in dealing with local patterns like discontinuities or sharp spikes **[10]**. However, in many applications, the signal of concern is both time and frequency dependent, and as such, the Fourier theory is ñincapableòof providing a global and complete analysis. The shortcomings of the Fourier transform, in addition to its failure to deal with non-periodic signals led to the adoption by the scientific community of a windowed version of this transform known as the STFT. The STFT transform of a signal f(t) is defined around a timeθ through the usage of a sliding window w (centred at time θ) and a frequency ω as:

$$STFT(\theta,\omega) = \int_{-\infty}^{+\infty} f(t)w(t-\theta)e^{-j\omega t}dt \qquad 2.3$$

As it is apparent from equation 2.3, even if the integral limits are infinite, the analysis is always limited to a portion of the signal, bounded by the limits [-θ,θ] of the sliding window. The time-frequency plane of a fixed window STFT transform is illustrated in Figure 2.1.

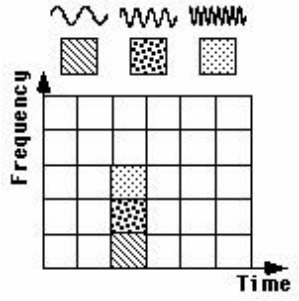


Figure 2.1: Fourier time-frequency plane **[10]**

Although, this approach (using STFT transform) succeeds well in giving both time and frequency information about a portion of the signal, however, as its predecessor, it has a major drawback. The fact is that the choice of the window size is crucial. As stated in **[13]**: ò the smaller the window size, the better the time-resolution. However, the smaller the window size also, the more the number of discrete frequencies which can be represented in the frequency domain will be reduced, and therefore the more weakened will be

the discrimination potential among frequencies". This problem is closely linked to the Heisenberg's uncertainty principle, which states that a signal (e.g. a very short portion of the signal) cannot be represented as a point in the time-frequency domain.

This shortcoming brings us to rise the fundamental question: how to size then the sliding window? Not surprisingly, the answer to this question leads us by means of certain transformations to the wavelet transform. In fact, by considering the convolution of the sliding window with the time-dependant exponential $e^{-j\omega t}$ within the integral of equation 2.3:

$$K_{\theta,\omega}(t) = w(t-\theta)e^{-j\omega t} \qquad 2.4$$

and replacing the frequency ω by a scaling factor a, and the window bound θ by a shifting factor b, leads us to the first step leading to the Continuous Wavelet Transform (CWT), as represented in equation 2.5.

$$K_{a,b}(t) = \frac{1}{\sqrt{a}}\psi^*\left(\frac{t-b}{a}\right) \qquad a \in R^+, b \in R \qquad 2.5$$

The combination of equation 2.5 with equation 2.3, leads to the CWT as defined by Morlet and Grossman **[14]**.

$$W(a,b) = \frac{1}{\sqrt{a}}\int_{-\infty}^{+\infty} f(t)\psi^*\left(\frac{t-b}{a}\right)dt \qquad 2.6$$

Where f(t) belongs to the square integrable functions space, $L^2(R)$. In the same way, the inverse CWT can be defined as:

$$f(t) = \frac{1}{C_\psi}\int_0^{+\infty}\int_{-\infty}^{+\infty}\frac{1}{\sqrt{a}}W(a,b)\psi\left(\frac{t-b}{a}\right)\frac{da\,db}{a^2} \qquad 2.7$$

The C_ψ factor is needed for reconstruction purposes. In fact, the reconstruction is only possible if this factor is defined. This requirement is known as the admissibility condition. In a more general way, $\psi(t)$ is replaced by $\aleph(t)$, allowing a variety of choices, which can enhance certain features for some particular applications **[13]**. However, the CWT in the form defined by equation 2.6 is highly redundant, which makes its direct implementation of minor interest. The time-frequency plane of a wavelet transformation is

illustrated in Figure 2.2. The differences with the STFT transform are visually clear.

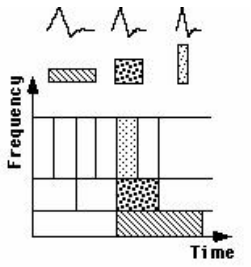


Figure 2.2: Wavelet time-frequency plane (**[10]** with modifications)

At this stage and after this brief introduction, it is natural to ask the question: therefore what are wavelet Transforms?

Although wavelet transforms are defined as a mathematical tool or technique, there is no consensus within the scientific community on a particular definition. As it has been stated by Sweldens in **[15]**: ñGiving that the wavelet field keeps growing, the definition of a wavelet continuously changes. Therefore it is impossible to rigorously define a waveletò According to the same author, to call a particular function a wavelet system, it has to fulfil the three following properties:

- Wavelets are building blocks for general functions: They are used to represent signals and more generally functions. In other words, a function is represented in the wavelet space by mean of infinite series of wavelets.
- Wavelets have space ï frequency localisation: Which means that most of the energy of a wavelet is confined in a finite interval and that the transform contains only frequencies from a certain frequency band.
- Wavelets support fast and efficient transform algorithms: This requirement is needed when implementing the transform.

Often wavelet transforms need O(n) operations, which means that the number of multiplications and additions follows linearly the length of the signal. This is a direct implication of the compactness property of the transform. However, more general wavelet transforms require O(nlogn) operations (e.g. undecimated wavelet).

To refine the wavelet definition, the three following characteristics have been added in **[16]** and **[8]**:

- Oneness of the generating function: Refers to the ability of generating a wavelet system from a single scaling function or wavelet function just by scaling and translating.
- Multiresolution ability: This concept, which has first been introduced by Mallat, states the ability of the transform to represent a signal or function at different level, by different weighted sums, derived from the original one.
- Ability of generating lower level coefficients from the higher level coefficients. This can be achieved trough the use of tree-like structured chain of filters called Filter Banks.

2.3 - Multiresolution

The multiresolution concept has been introduced first by Mallat **[9]**. It defines clearly the relationships between the QMF, pyramid algorithms and orthonormal wavelet bases through basically, the definition of a set of nested subspaces and a so-called scaling function. The strength of multiresolution lies in its ability to decompose a signal in finer and finer details. Most importantly, it allows the description of a signal in terms of time-frequency or time-scale analysis.

2.3.1- Nested Subspaces

The basic requirement for multiresolution analysis is the existence of a set of approximation subspaces of L^2 (R) (square integrable function space) with different resolutions, as represented schematically for the three intermediate subspaces in Figure 2.3 and stated by equation 2.8:

$$V_{-\infty} \ldots \subset V_{-1} \subset V_0 \subset V_1 \subset \ldots V_{\infty} = L^2(R) \qquad 2.8$$

In such a way that, if $f(t) \in V_j$ then $f(2t) \in V_{j+1}$. Which means that the subspace containing high resolution will automatically contains those of lower resolution. In a more general case, if $f(t) \in V_0$, then $f(2^k t) \in V_k$. This implication is known as the scale invariance property.

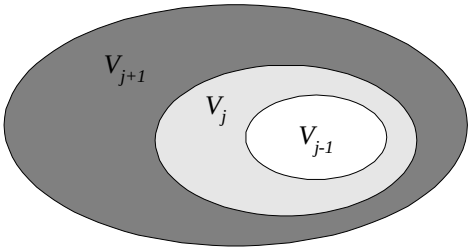


Figure 2.3: Nested subspaces

2.3.2 - Scaling Function

The existence of a so-called scaling function $\phi(t)$ is primordial in order to benefit from the multiresolution concept. In this context, let us define the scaling function first and then define the wavelet function through it **[8]**. Let the scaling function be defined by the following equation:

$$\phi_k(t) = \phi(t-k) \qquad k \in Z \qquad \phi \in L^2(R) \tag{2.9}$$

which forms with its translates an orthonormal (The orthogonality is not necessary, since a non orthogonal basis (with the shift property) can always be orthogonalised **[8]**) basis of the space V_0:

$$V_0 = \underset{k}{span}\{\phi_k(t)\} \tag{2.10}$$

This means that any function belonging to this space ($f(t) \in V_0$) can be expressed as a linear combination of a set of so-called expansion coefficients, with the scaling function and its consecutive translates (since $\phi_k(t)$ are the basis functions):

$$f(t) = \sum_k c_k \phi_k(t) = \sum_k c(k)\phi(t-k) \tag{2.11}$$

where the expansion coefficients c_k (or $c(k)$) are calculated using the inner product:

$$c_k = < f(t), \phi_k(t) > \tag{2.12}$$

By simply scaling and translating, a two-dimensional scaling function is generated from the original scaling function defined in equation 2.9:

$$\phi_{j,k}(t) = \frac{1}{\sqrt{a}}\phi\left(\frac{t-b}{a}\right) \tag{2.13}$$

Where a and b are, the scaling and the shifting factors as defined in 2.5, respectively. To ease the implementation of a wavelet system, the translation and the scaling factor have been adopted to be a factor of two **[10]**. In fact:

$$a = 2^{-j} \quad , \quad b = 2^{-j}.k \tag{2.14}$$

These values are adopted for the remaining of the chapter. Thus, equation 2.13 can be rewritten as:

$$\phi_{j,k}(t) = 2^{j/2}\,\phi(2^j t - k) \tag{2.15}$$

Identically, the two-dimensional scaling function forms with its translates an orthonormal space over k:

$$V_j = \underset{k}{span}\{\phi_{j,k}(t)\} \qquad k \in Z \quad and \quad j \in Z \tag{2.16}$$

And as such any function $f(t)$ of this space can be expressed as:

$$f(t) = \sum_k c(k)\phi(2^j t + k) \tag{2.17}$$

As a consequence, if $\phi(t) \in V_0$, then since $V_0 \subset V_1$, $\phi(t)$ can be expressed as a linear combination of the scaling function $\phi(2t)$ spanning the space V_1:

$$\phi(t) = \sum_k h(k)\sqrt{2}\phi(2t - k) \tag{2.18}$$

where the coefficients h(k) are the scaling function coefficients. The value $\sqrt{2}$ ensures that the norm of the scaling function is always equal to the unity. This equation is fundamental to the multiresolution theory and is called the multiresolution analysis equation.

2.4 - Wavelet Function

What has been done so far to define the scaling function, its translates and the corresponding spanned spaces, can also be applied in the same way to the so-called wavelet function. Let us suppose for this purpose that the subspace $V_0 \subset V_1$ has an orthogonal complement W_0, such as V_1 can be represented as a combination of V_0 and W_0 as follows:

$$V_1 = V_0 \oplus W_0 \tag{2.19}$$

where the complementary space W_0 is spanned also by an orthonormal basis:

$$\psi_k(t) = \psi(t-k) \qquad k \in Z \qquad \psi \in L^2(R) \tag{2.20}$$

The function $\psi(t)$ is known as the mother wavelet, the wavelet prototype or the wavelet function. The same properties, which apply to the scaling function, are also applicable to the wavelet function. In other words, a function $f(t) \in W_0$ can be expressed as:

$$f(t) = \sum_k d_k \psi_k(t) = \sum_k d(k) \psi(t-k) \tag{2.21}$$

where, the expansion coefficients d_k (or $d(k)$) are calculated using the inner product:

$$d_k = < f(t), \psi_k(t) > \tag{2.22}$$

Likewise, since $W_0 \subset V_1$, $\psi(t)$ can also be expressed in terms of the scaling function $\phi(2t)$ of the higher space V_1:

$$\psi(t) = \sum_k g(k) \sqrt{2} \phi(2t-k) \tag{2.23}$$

where $g(k)$ are the wavelet coefficients. This leads to a dyadic decomposition as represented by the grid of Figure 2.5. The equation 2.19 can be generalised to an arbitrary number of subspaces, such as, V_2 is represented in terms of V_1 and W_1, V_3 in terms of V_2 and W_2, and so on. The whole decomposition process is illustrated in Figure 2.4.

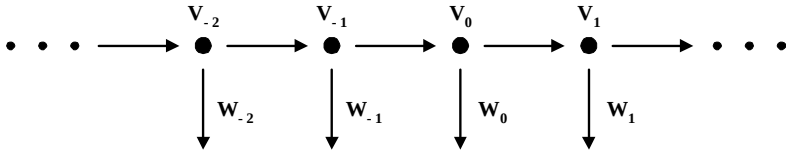


Figure 2.4: Space decomposition

More generally, a subspace V_j is spanned by W_{j-1} and V_{j-1}. Thus, the $L^2(R)$ space can be decomposed as follows:

$$L^2(R) = V_j \oplus W_j \oplus W_{j+1} \oplus W_{j+2} \oplus ....W_0 \oplus W_1.... \tag{2.24}$$

The index j represents the depth or the level of decomposition, which is arbitrary in this case. As for the scaling function, a two-dimensional scaled and translated wavelet function is defined as:

$$\psi_{j,k}(t) = 2^{j/2}\psi(2^j t - k) \quad 2.25$$

in such way that:

$$W_j = \underset{k}{span}\{\psi_{ji,k}(t)\} \quad 2.26$$

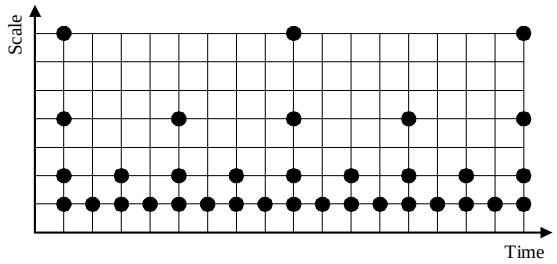


Figure 2.5: Dyadic wavelet transform space representation

2.5 - Series Expansions and Discrete Wavelet Transforms

According to equation 2.24, a function $f(t)$ belonging to the $L^2(R)$ space can be expanded in series in terms of the scaling function spanning the space V_j and the wavelet functions spanning the spaces $W_j, W_{j+1}, W_{j+2}, ..., W_0, W_1....$ as follows:

$$f(t) = \sum_k c_j(k)\phi_{j,k}(t) + \sum_{n=j}^{+\infty}\sum_{k=-\infty}^{+\infty} d_n(k)\psi_{n.k}(t) \quad 2.27$$

where $\phi_{j,k}(t)$ is defined by equation 2.15 and $\psi_{n.k}(t)$ is defined by equation 2.25. In this case, the index j, which is arbitrary, represents the coarsest scale, while the remaining are the high resolution details. Equation 2.27 represents the wavelet expansion series of the function $f(t)$, which plays a major role when deriving a more practical form of the wavelet transform.

The coefficients in the wavelet expansion series $c_j(k)$ and $d_n(k)$ (or $c(j,k)$ and $d(n,k)$) are the so-called discerete wavelet transform of the function $f(t)$. Since the basis functions are orthonormal, they can be calculated using equations 2.12 and 2.22, respectively. We will see later in this chapter that the orthonormality condition can be relaxed allowing the implementation of another important basis, namely, the biorthogonal basis.

2.6 - Filter Banks and Wavelet Implementations

In general, wavelet transform-based applications involve discrete coefficients instead of scaling and/or wavelet functions. For practical and computational reasons, discrete time filter banks are required. Such structures decompose a signal into a coarse representation along with added details. To achieve this representation, the relationship between the expansion coefficients at lower and higher scale levels need to be defined. This can be easily done by using a scaled and shifted version of equation 2.18 along with simple transformations as reported in **[8]**. This relation is defined by:

$$c_j(k) = \sum_n h(n-2k)c_{j+1}(n) \qquad 2.28$$

and

$$d_j(k) = \sum_n g(n-2k)c_{j+1}(n) \qquad 2.29$$

where $n \in Z$ and $k \in Z$. The computation of such equations is achieved through the use of the well-established digital filtering theory. In particular, for finite length signals (which is the case for digital images), the use of a Finite Impulse Response filter (FIR) is the most appropriate choice. However, since equations 2.28 and 2.29 compute one output for each two consecutive inputs, a modification needs to be made. The basic operation required here, is derived from the multirate signal processing theory **[17]**. It simply consists of using a down-sampler or decimator by a factor of two. In practice, it consists of applying a pair of FIR filters; each followed by a decimator as illustrated by Figure 2.5:

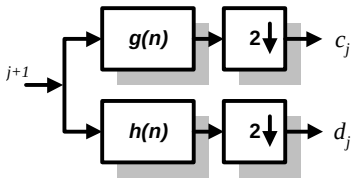


Figure 2.6: Analysis Filter Bank

The filter bank is defined as a combination of a low pass filter and high pass filter, both followed by a factor of two decimation **[18]**. Thus, the decomposition is reduced to two basic operations from the digital signal processing theory: a filtering and a down sampling.

The structure in Figure 2.6 is generally used to implement Mallat's algorithm. To allow further level of decomposition, identical stages are cascaded leading to a multiresolution analysis. This analysis scheme is known as the Subband Coding structure **[8]** and is illustrated in Figure 2.7.

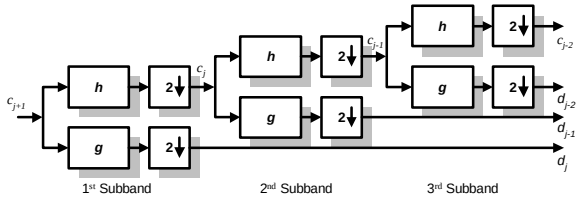


Figure 2.7: Three-Stage analysis Subband Coding

At each stage, the spectrum frequency of the signal analysed is halved by a factor of two. This leads to a logarithmic set of bandwidths as illustrated by Figure 2.8.

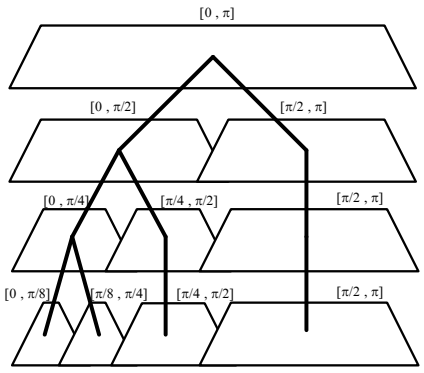


Figure 2.8: Frequency Spectrum of a three-stage Subband Structure

To recover the original signal from the previously analysed one, a reversed version of the analysis filter bank of Figure 2.6 is required. This can be achieved by using two basic operations: a filtering and an up sampling or interpolating process. In multirate digital signal processing, appending a zero sample between two consecutive samples performs the up sampling. Thus, for each input sample, we get two output samples. A three stage synthesis subband coding is illustrated in Figure 2.9.

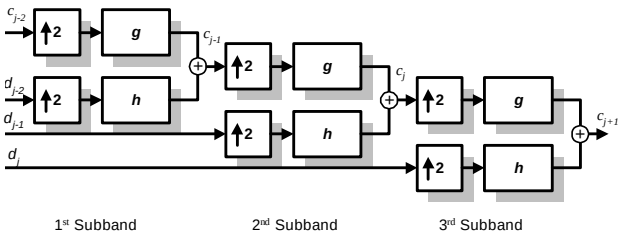


Figure 2.9: Three-Stage synthesis Subband Coding

2.7 - Wavelet Transform Computation

We focus on both 1-Dimensional and 2-Dimensional systems. The algorithm dedicated to implement the wavelet transform is Mallatôs Pyramidal algorithm. Which is a direct consequence of the multiresolution concept Up to date, it is the most widely used approach-both in software and hardware - for implementing the wavelet transform **[19]**. Since the one-dimensional decomposition and reconstruction schemes have been already introduced in section 2.6, we will focus in this section on two-dimensional schemes, which are more suitable for image analysis and synthesis. The two-dimensional decomposition approach is based on the property of separation of the functions into arbitrary x and y directions. The first step is identical to the one-dimensional approach; however, instead of keeping the low-level resolution and processing the high level resolution, both are processed using two identical filter banks after a transposition of the incoming data. Thus, the image is scanned in both horizontal and vertical directions. This result in an average image (or subimage) and three detail images generated by the following 2-D scaling function $\phi(x,y) = \phi(x)\phi(y)$ and the vertical, the horizontal and the diagonal wavelet functions: $\psi_1(x,y) = \phi(x)\psi(y)$, $\psi_2(x,y) = \psi(x)\phi(y)$ and $\psi_3(x,y) = \psi(x)\psi(y)$, respectively. To recover the original image, the inverse process is applied. Figure 2.10 illustrates the analysis and synthesis stages built using three filter banks each.

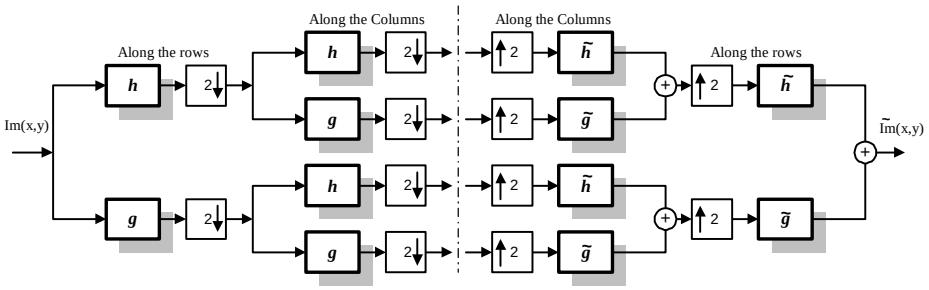


Figure 2.10: Two-dimensional Mallat's analysis and synthesis tree

In this case, the frequency band is halved at each stage by a factor of four as represented by Figure 2.11.

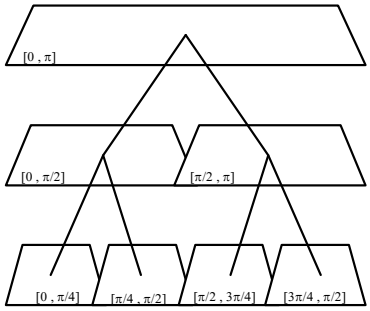


Figure 2.11: Frequency Bands of Mallat's 2-D Analysis Algorithm

2.8 - Practical considerations of Wavelet Transform:

In many practical problems, both the orthonormal basis and the biorthogonal basis can be used. The two bases (or families) present similarities and differences. Another scheme, called wavelet packet, which involves either orthonormal basis or biorthogonal basis is also possible. The following briefly describes the main features of orthonormal and biorthogonal bases together with extension to the wavelet packet scheme. It is worth mentioning that other schemes like undecimated wavelet, adaptive wavelets and multiwavelets exist and are beyond the scope of this brief overview.

2.8.1 - Orthonormal Basis

The orthonormal basis emerged from the work initiated by Mallat and Daubechies **[9][16]**. The orthonormality property is somewhat seen as the discrete version of the orthogonality property **[19]**. However, the basis functions are further normalised. These concepts have been mentioned when the multiresolution feature and the scaling function have been introduced. The admissibility and the orthogonality conditions ensure the existence and the

orthogonality feature of the scaling function, defined by equation 2.18. This is achieved if:

$$\sum_n h(n) = \sqrt{2} \qquad 2.31$$

and

$$\sum_n h(n)h(n+2k) = \delta(k) \qquad 2.32$$

Furthermore, using the two equations above alongside with equation 2.23, which defines the wavelet function, the orthogonality of the scaling function and the wavelet function at the same scale can be derived. This can be achieved only if the following equality is verified:

$$g(n) = (-1)^n h(1-n) \qquad 2.33$$

The orthogonality between the wavelet coefficients and the scaling coefficients is then only a simple implication:

$$\sum_n h(n)g(n) = 0 \qquad 2.34$$

The scaling coefficients, which satisfy equation 2.32, are called Quadrature Mirror Filters (QMF). To achieve perfect reconstruction, the analysed signal has to be identical to the synthesised one. In other words, $c_j(n) = \tilde{c}_j(n)$, where $c_j(n)$ and $\tilde{c}_j(n)$ are the input and the output of a two-band filter (or filter bank) as shown in Figure 2.12, respectively.

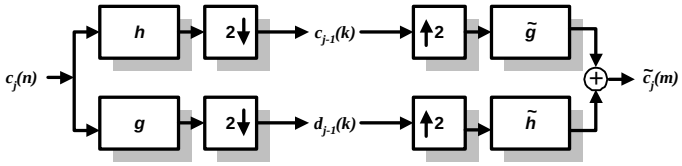


Figure 2.12: Two-band analysis and synthesis filter bank

2.8.2 - Biorthogonal basis

Biorthogonal wavelet basis can be seen as a generalisation of the orthogonal wavelet basis where some imposed restrictions on the latter have been relaxed. Unlike the case of orthogonal basis, the scaling and the wavelet functions need be neither of the same length, nor even numbered. Hence, the quadrature mirror property is not applicable and is replaced with a dual

property. For the perfect reconstruction equation to hold, the scaling and the wavelet coefficients have to fulfil the following equations:

$$\tilde{g}(n) = (-1)^n h(1-n) \qquad 2.35$$

and

$$g(n) = (-1)^n \tilde{h}(1-n) \qquad 2.36$$

It is clear that when the analysis and the synthesis filters are similar, the system becomes orthogonal. The ñorthogonalityò condition in this case is defined by:

$$\sum_n \tilde{h}(n) h(n+2k) = \delta(k) \qquad 2.37$$

Previously, in orthogonal basis, only the analysis scaling coefficients (or wavelet coefficients) along with their shifted versions were used. In biorthogonal case, the analysing scaling coefficients are kept unchanged, while their shifted versions are replaced by the shifted versions of the synthesis dual filter. In other words, the analysis filter is orthogonal to its synthesis dual filter. The biorthogonal denomination comes from this feature.

At the expense of the energy partitioning property stated by Perseval's equality, which is a direct consequence of the lack of orthogonality, a greater flexibility can be achieved by using the basis and dual basis **[8]**. One of the most ñimportantò features in the biorthogonal basis is the linear phase property, which leads to the filter coefficients (when implementing a wavelet system) being symmetric. In addition, the difference of length between dual filters must be even, leading either to odd or even length of the low pass and the high pass filters. In general, biorthogonal wavelet systems present the following features:

- The coefficients of the filters are either real numbers or integers.
- The filters in this family present either even or odd orders.
- The low pass and the high pass filters used in the filter bank have not the same length.
- The low pass filter is always symmetric.
- The high pass filter is either symmetric or antisymmetric.

2.8.3 - Wavelet Packets

In contrast to the ñtraditionalò Mallatôs decomposition, which leads to narrow frequency bandwidths (low frequencies) and wide frequency bandwidths (high frequencies), the wavelet packet approach emerged first as a way of adjusting high frequency resolutions. Hence, the Mallatôs decomposition scheme is applied to both parts of a filter bank leading to the split of frequencies in progressive finer resolutions. The generic structure of wavelet packet decomposition is shown in Figure 2.19 the frequency bandwidths illustrated by Figure 2.20. In this scheme, the number of filters increases by a factor of $(2^i - 2^j)$ at each successive subband, where i and j represent two consecutive resolutions and $i - j = 1$.

In comparison to classical wavelet approach, the wavelet packet scheme presents the following features **[16]**:

- Possibility of using different wavelets from a level to another. This strategy has been used in **[19]** to implement a two-level orthonormal wavelet packet and a three-level biorthogonal wavelet packet.
- Possibility of choosing particular wavelet packet decomposition from the general generic structure of Figure 2.13. Thus, one can choose either to preserve the orthonormality feature of the decomposition, or highlight the peculiarities of the signal **[19]**.

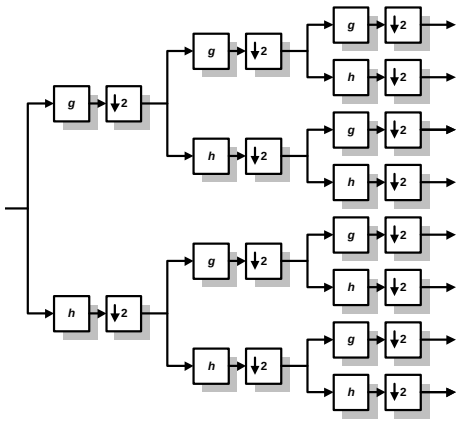


Figure 2.13: Three-stage Wavelet Packet Decomposition

However, there is a cost to be paid. In this case, the computational complexity of a wavelet packet structure is $O(n\log(n))$ in contrast to the $O(n)$ of the classical wavelet transform.

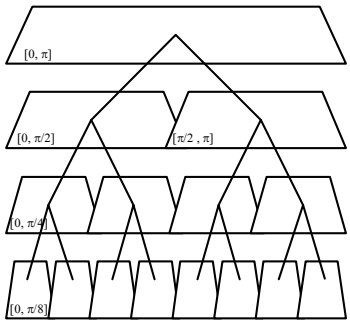


Figure 2.14: Two-band analysis and synthesis filter bank

2.9 - Wavelet-based Applications

Recently, The wavelet transform is being increasingly used, not only in the field of image and signal processing applications but also in many other different areas, ranging from mathematics, physics, astronomy to statistics and economics **[8][18]**. In image processing based applications, image compression, image denoising and image watermarking are at the cutting edge, and as such, a brief description of these wavelet-based applications is given in the following subsections.

2.9.1- Image compression

Even though the wavelet transforms have been widely used in image coding since the late 80s, they only gained their notoriety in the field by the adoption of the first wavelet-based compression standard scheme, known as the FBI fingerprint compression standard. Recently, what did Sweldens state in **[15]** as a need of standardising a wavelet-based compression scheme under the header ñproblems not sufficiently explored with waveletsò, has seen the day, by the adoption of the JPEG2000 new compression standard **[12]**. The block diagram of the JPEG2000 standard does not really differ from the JPEG standard one. The discrete wavelet transform, which replaces the DCT, is applied first to the source image. The transformed coefficients are then quantised. Finally, the output coefficients from the quantiser are encoded (using either Huffman coding or arithmetic coding techniques) to generate the

compressed image. To recover the original image the inverse process is applied. Figure 2.15 shows the basic JPEG2000 Encoding Scheme **[12]**.

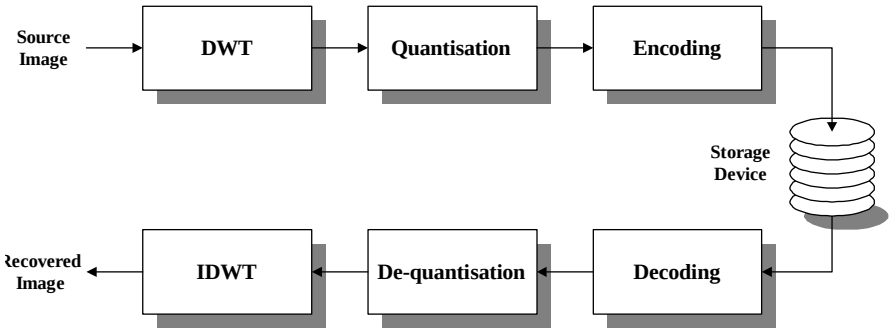


Figure 2.15 Wavelet-based encoding scheme

2.9.2 - Image Denoising

Image manipulation, includes a wide range of operations like digitising, copying, transmitting, displaying. . etc. Unfortunately, such manipulations generally degrade the image quality by spanning many types of noise. Hence, to recover the original structure of the image, the undesired added noise needs to be localised and then removed. In image processing, noise removal is achieved through the usage of filtering-based denoising techniques. Traditionally, image denoising or image enhancement is performed using either linear filtering or non-linear filtering. Linear filtering is achieved either by using spatial techniques, as low pass filtering, or frequency techniques, as the Fast Fourier Transform (FFT). On the other hand, statistical and morphological filters are typical examples of non-linear filtering. However, the filtering techniques lead in some cases to baneful effects when applied indiscriminately to an image. In fact, if it is not the whole image that is blurred, some of its important features (e.g. edges) are.

A solution to overcome this problem has been introduced by Denoho and Johnstone **[20]**. Instead of exploiting either linear or non-linear filtering, their technique consists of using the DWT followed by a thresholding operation. This method exploits the energy compaction ability of the wavelet transform to separate the image from the added noise. The role of the threshold is to eliminate the noise present in the image. Finally, the enhanced/ódenoisedò

image is recovered by applying the inverse DWT. This method is also known as the wavelet shrinkage denoising, and is classified as a nonlinear processing technique due to the thresholding operation involved in the process as illustrated in Figure 2.16.

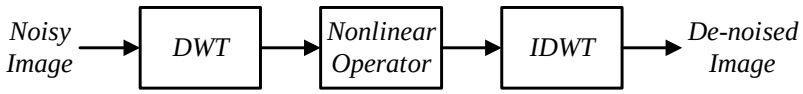


Figure 2.16: Wavelet-based denoising system

2.9.3 - Image Watermarking **[21]**:

Image watermarking emerged in the mid 90s as a discipline, among the wide range of multidisciplinary field of data hiding, as a methodology of protecting digital images from any piracy act. It consists of embedding a watermark (a trace) within a digital image before using or publishing it. The efficiency of a watermarking method lies generally in its ability to fulfil three requirements: robustness, security and invisibility.

Watermarking techniques can be classified into two categories; spatial domain methods and transform-based methods. The wavelet-based watermarking technique falls into the latter. The wavelet facilitates a simultaneous spatial localisation and frequency spread of the watermark within the source image. It has been shown that the method is robust under compression, additive noise and filtering **[21]**.

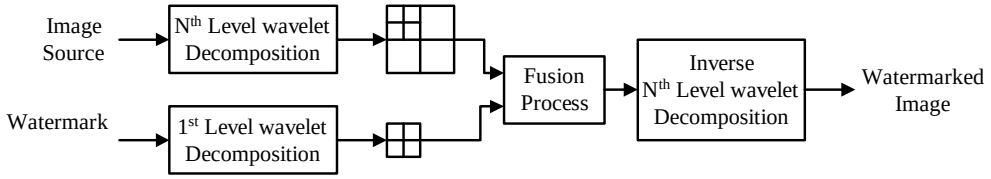


Figure 2.17: Wavelet-based watermarking system

2.10 - summary

Since the late 80s, the wavelet transform has been widely used in different scientific applications including signal and image processing. This ongoing growing success, which has been characterised by the adoption of some wavelet-based schemes, is due to features inherent to the transform, such as time-scale localisation and multiresolution capabilities. In this chapter, the

basic concepts of the wavelet transform have been introduced. First, the historical development of the wavelet transform and its advent to the field of signal and image processing were reviewed. Then, its features and the mathematical foundations behind it were reviewed. To ease the understanding of the wavelet theory, the related notations and terms, such as the scaling function, multiresolution, filter bank and others were described and then briefly explained.

Depending on the application at hand, Mallatôs algorithm for implementing the wavelet transform have been developed. Finally, some wavelet based image processing applications were also given.

CHAPTER 3

FIELD PROGRAMMABLE GATE ARRAY

3.1 - Introduction

Versatile hardware, can perform a wide range of operations and tasks, but fails to reach the system speed of a more specialised hardware. On the contrary, an oriented application-specific hardware, such as Application Specific Integrated Circuits (ASICs), can perform a restricted set of operations/tasks more quickly. Hence, configurable computing, generally in the form of Field Programmable Gate Arrays (FPGAs), appears to be the promising land for hardware designers. This old/new paradigm allies the flexibility of software while preserving the hardware performances. Unlike the case of custom hardware in the form of ASICs, which cannot be reused for a slightly different problem to the one they were designed for, configurable hardware based, FPGAs allows modifications at almost any stage of the design process. In fact, configurable hardware is easily upgraded (due to its inherent nature) to suit any changes of a primal design. Used in a desktop, configurable hardware can be tailored to speed up or accelerate applications, which require a system speed superior to the one offered by general purpose processors. The hardware here needs to adapt itself to continual changes in response to end users needs. Obviously, the configurable capabilities of such hardware will not eliminate the need for general-purpose microprocessors running on today's Personal Computers (PCs). In fact, "FPGAs will never replace microprocessors for general-purpose computing tasks", as stated in **[22]**.The idea of configurable or reconfigurable computing arose first at the late 60s at the University of California at Los Angeles (UCLA) **[22][23]**. However, the real emergence of this new paradigm for hardware computation was piloted by the commercialisation of the first SRAM-based FPGA by Xilinx Corporation in 1986. The first configurable devices from both Xilinx Corporation **[6]** and Altera Corporation **[24]**, composed typically of a fine grained structure, allowed a system speed in the range of 2MHz – 5MHz and a chip area of less than a 100 of logic blocks. The efforts deployed by

scientists since then brought to light new developments but also new challenges. The developments in the microelectronic technology led to the emergence of a new range of devices providing a system gate beyond a million (e.g. Xilinx Virtex family) and to the continual emergence of a wide range of FPGA based systems. The flexibility of these systems has also been improved by the development of high level tools for automation, design capture, synthesis, optimisation, mapping, and rapid generation.

This chapter does not claim to give a study about field programmable devices based reconfigurable systems and their fields of application, but instead, gives a brief overview, which we found necessary as a part of this thesis. In this context, we will briefly introduce different FPGA devices from different manufacturers with a special emphasis on XilinxTM material (since this has been adopted throughout this work). Finally, we will focus on some FPGA-based wavelet applications by highlighting their features and characteristics and then by assessing their performances.

3.2 - Prototyping, Configuration and Reconfiguration:

As part of a digital design process, FPGA devices play different roles depending on the end-user wills and goals. However, there is still confusion on certain terms used in the field such as prototyping, configuring and reconfiguring. Commonly, FPGA devices are used for prototyping purposes in an ASIC design process. In this case, FPGAs are used as prototype hardware at the early stages of the design and replaced by an ASIC during the production. The fact is that an FPGA is a low cost hardware for small and medium series of production, however, when it comes to larger series, ASIC is so far a better solution. Generally, the FPGA reconfigurable computation capability is not involved in a prototyping process.

Reconfigurable computing arises when FPGAs' run-time capability is involved. In this case, the design implemented may change in response to an external request while the system is still running. In other words, the FPGA acts as an execution engine for different hardware functions and as such can be called, by analogy to a CPU, a Reconfigurable Processing Unit (RPU) **[23]**. Using the run-time capability, the FPGA can reconfigure itself to run a succession of

tasks on the same hardware. In this situation, the chip operates in a time-sharing mode by swapping rapidly from one task to another **[22]**. Hence, the system designer is able to execute more hardware than available. This is reconfigurable computing at its best, and using this approach it is possible to design systems that do more, cost less, and have shorter design and implementation cycles **[23]**. Obviously, to appreciate the benefits of reconfigurable computing, FPGAs with fast “re”-programming time are required.

Configurable computing can be seen as a step between the two. In this scheme, the logic within the FPGA device is changed when necessary. A possible use is for hardware upgrade. In this case, the internal logic is redesigned and further downloaded to reconfigure the chip. The same configuration file can be sent to customers to upgrade the hardware in the same way they do with software. The configuration capability of an FPGA is also involved in the prototyping process since the device is continually configured until the correct functionality of the design is achieved.

3.3 - Field Programmable Gate Arrays:

The basic architecture of an FPGA consists of a two dimensional array of cells. Different types of interconnection resources provide the internal communications between cells in this matrix-like structure. At the edges surrounding the whole device, Input Output Blocks (IOBs) interface the chip with the external world. The architecture of a typical FPGA is illustrated in Figure 3.1.

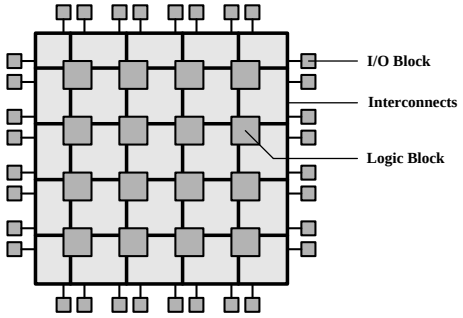


Figure 3.1: Typical Architecture of Field Programmable Devices

3.3.1- Xilinx Virtex II FPGA Series:

The Xilinx Virtex II family of FPGAs provides a regular, flexible, programmable architecture of Configurable Logic Blocks (CLBs) interconnected by a hierarchy of versatile routing resources and surrounded by a perimeter of programmable IOBs **[6]**. The devices are customised by loading configuration data into the internal static memory blocks (SRAMs).

The CLBs provide functional elements for constructing user's logic. IOBs provide the interface between the package pins and internal signal wires. The programmable interconnect resources provide routing paths to connect the inputs and the outputs of the CLBs and IOBs to the appropriate network. Customised configuration is provided by programming internal static memory blocks that determine the logic functions and interconnection in the Logic Cell Array (LCA) device.

3.3.1.1 - Configurable Logic Block

Due to its structure, a Xilinx Virtex II FPGA CLB can either implement a pure combinatorial logic system, a pure sequential logic system or both types. The CLB is composed of two 4-inputs function generators (slices) implemented as memory Look Up Tables (LUT), a 3-input function generator and 2 storage elements. Figure 3.2 illustrates the general structure of a CLB.

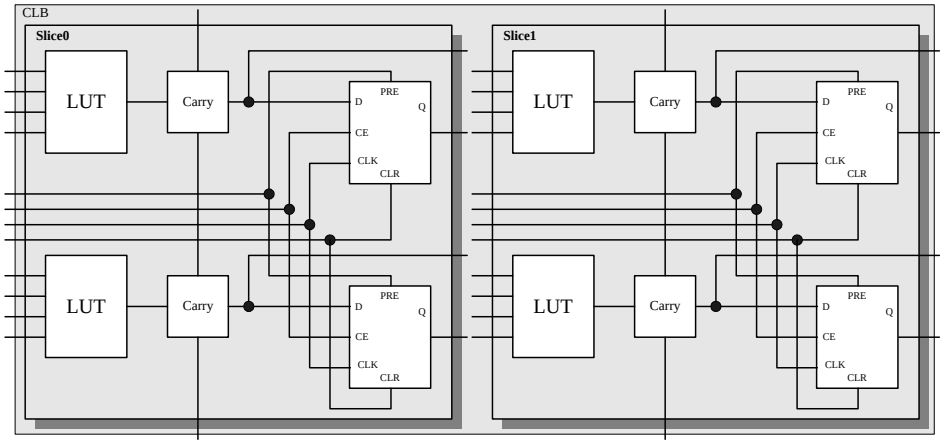


Figure 3.2: Simplified Virtex II CLB Structure

The 4-input function generators, labelled G and F, are driven by four independent inputs, which can implement any arbitrarily Boolean function. The 3-input function generator, labelled H, is driven by the outputs of G and F function generators, while the remaining input comes from outside the CLB.

Consequently, by combining either two or three function generators, a CLB can implement any arbitrarily Boolean function from five and up to nine inputs (see Figure 3.3). On the other hand, a CLB presents four outputs, two of each category, which, depending on the application, are registered or non-registered. The storage elements can be configured either as latches or flip flops with Clock Enable (EC) and asynchronous Set/Reset (S/R).

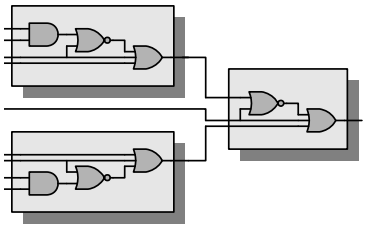


Figure 3.3: Implementation of a 9-inputs Boolean Function into a CLB Function Generators

3.3.1.2 - Input Output Blocks

Surrounding the whole chip, user programmable IOBs provide the interface between the internal logic and the external package pins. Input signals are brought to the internal array through two lines I1 and I2. These two inputs can be routed through either a storage element, which can be configured either as an edge triggered flip-flop or a transparent latch, or passed directly to the chip. Additional programmable resources, such as programmable pull-up and pull-down resistors can be used to separate input and output clock signals and Global Set/Reset signals. Figure 2.4 illustrates the typical structure of an IOB.

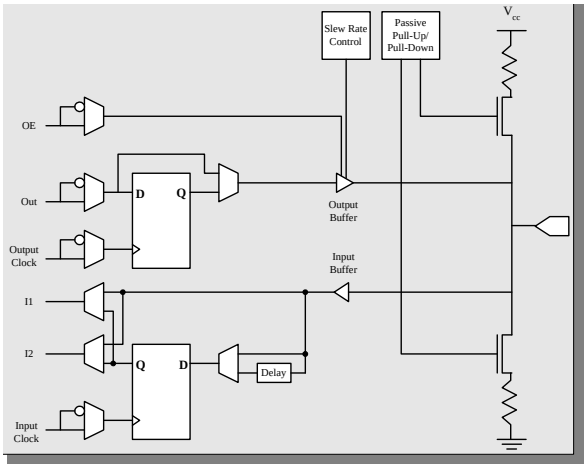


Figure 3.4: IOB Internal Structure

An interesting feature of the IOBs arises when input signals are passed through a storage element and in the same time input directly to the chip. This capability allows the IOB to demultiplex external signals such address/data buses by storing data within the storage element (e.g. configured as flip-flop) and passing data into the wiring. Using inputs to drive wide decoders built into the wiring for fast address recognition is another IOB potential, which we found worth to mention in this brief description. In fact, this IOB capability eases bus interfacing.

3.3.1.3 - Routing Resources

Routing within the FPGA chip is implemented as programmable switching points through the use of metal segments. As the FPGA CLB inputs and outputs are distributed on each side (north, south, west and east), it is then natural that routing resources are surrounding it as it is clearly illustrated in Figure 3.5. The number of routing channels depends mainly on the FPGA size. They are divided in three main types depending on the length of their segments. Single Length Lines: Basically a grid of horizontal and vertical lines used locally to drive signals and connect nets with fanout superior to one. The single length lines intersect with Switch Matrices to provide the appropriate connections.

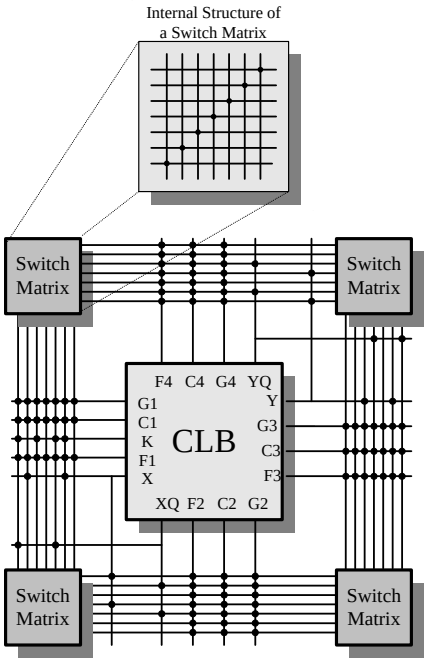


Figure 3.5: CLB Connections to Single Length Lines

Double Length Lines: Grouped in pairs with the Switch Matrix in a way that each line goes through it at every other CLB location in that row/column. Figure 3.6 illustrates double long lines routing distribution in a FPGA chip.

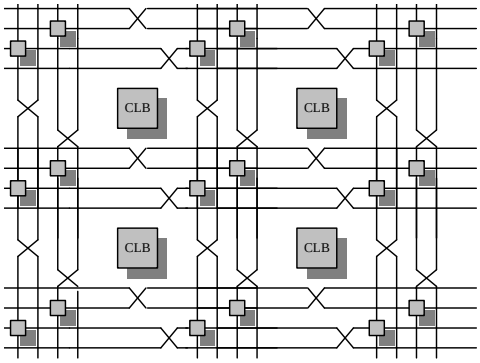


Figure 3.6: Double Length Lines Routing Resources

Long lines: They form a grid of interconnect segments that run entirely the length/width of the chip. In addition, special global buffers used for fast clock distribution at minimal skew (and other high fanout control signals) drive additional long line routing resources. A pair of three state buffers surrounding the CLB is used to route CLB outputs to the long lines. Another alternative is the uses of single interconnect length lines. Long line routing distribution is illustrated in Figure 3.7.

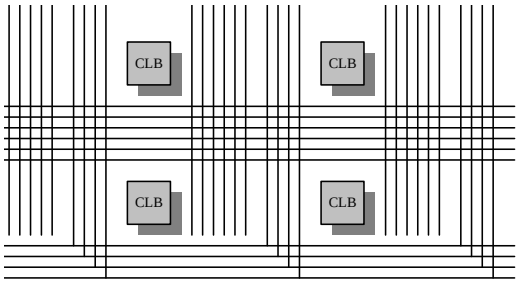


Figure 3.7: Long Line Routing Resources

Communication between long lines and single length lines take place in the programmable interconnect points at the line intersections. However, double length lines cannot be connected to any other lines.

Furthermore, each CLB nests into a Versatile Block, VersaBlockTM, which provides local routing resources to connect the CLB to the GRM **[6]**. An inner Versatile Ring, VersaRingTM, input/output interface provides additional routing resources, which improves routability and eases pin locking. In addition, the Virtex structure provides other built-in resources such as the

dedicated block memories and the clock DLLs that connect to the GRM. Each memory Block can hold up to 4096 bits. The dedicated clocks are used for clock-distribution delay compensation and clock domain control. The Virtex FPGA also supports the fast carry logic capability; however, this built in resource is implemented differently in comparison with the 4000 FPGAs.

3.3.3 - Other FPGAs :

Even though Xilinx Corporation shares a considerable part of the FPGA market, others FPGA manufacturers also provide different type of FPGA devices. For example, FPGAs are provided by Altera corporation (such as APEX II, APEX and FLEX EPF 8k, 10k and 20k series), Lucent/AT&T (such as ORCA 1, 2 & 3 series) and Actel (such as ProASIC A500k, Actel MX and Actel SX) **[24][25]**. Different FPGA architectures from different corporations is beyond the scope of this thesis.

3.4 - Design Environment :

Configuring FPGA chips is not a straightforward operation. Normally, it involves a number of different steps, in which different tools can be used.

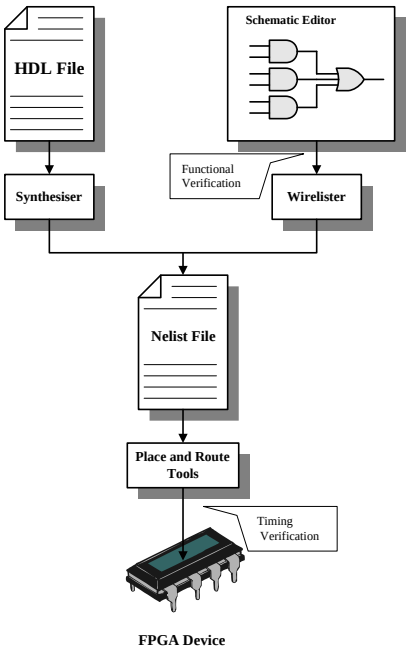


Figure 3.8: Simplified FPGA Design Flow **[26]**

The process starts by choosing a design capture tool, generally, either a hardware description language or a schematic tool. The next step consists of converting the hardware description file or the schematic design into a netlist

format. This is achieved through the use of synthesisers for hardware description files and wirelisters for schematic captures. To generate the bit stream necessary for configuring the FPGA, the netlist file is passed through implementation tools (e.g. map, place and route tools). A simplified FPGA design process is illustrated in Figure 3.8.

Generally, the design flow incorporates a verification process at different levels. Functional verification checks the correctness of the design at functional level only without taking into account any propagation on chip delays. It represents the first level of verification. The second level of verification takes into account different type of delays within the chip to build a timing model. This process is called timing analysis, which is the basis of the so-called timing verification. When finally the FPGA is configured, the last type of verification is carried out.

3.4.1- Design entry

Digital designs are entered using Schematic captures, Truth tables, state machines and Hardware Description Languages (HDLs). To benefit from each tool, a combination between them is always possible. In what follows, a brief overview of the two most commonly used capture tools is presented.

3.4.1.1 - Schematic Capture tools

Schematic capture tools have been widely used in the early days of FPGAs. They allow a designer to specify a digital circuit as a 2-D diagram in which elementary logic components and logic macros representing different level of abstraction are connected together using different routing resources. A number of front-end design tools, such as the ones provided by View Logic and Mentor Graphics, can be used for this purpose.

Generally, libraries of basic logic components are added to Schematic capture environment. These libraries alongside with different other varieties of resources, typically in the form of core generators (such Xilinx CoreGenTM), are supplied by FPGA constructors.

Consequently, a design dedicated to a particular FPGA architecture cannot be used for configuring another architecture, especially when targeting devices

from different foundries. In addition, once the design is converted into a netlist format, all the topological information is lost.

Schematic capture tools are used to design small circuits. For larger circuits, their use can be tedious and time consuming. In such cases, the schematic capture becomes more an obstacle than a helping tool. This is where hardware description languages enter the scene.

3.4.1.2 - Hardware Description Languages

Nowadays, HDLs are becoming widely used to capture digital designs. In parallel, Electronic Design Automation (EDA) tools have also been keeping growing at an incredible rate. This rapid growth has led to the emergence of faster and more efficient ways to design digital circuits. Therefore, this is achieved through the usage of HDLs.

Although, there are many HDLs available in the market, VHDL (Very high speed integrated circuit Hardware Description Language) **[27]** and Verilog® **[28]** remain so far the most commonly used languages. HDLs look like conventional high level programming languages (C, Java.), and as such, textual editors are used to capture them. These languages allow the description of a design using different paradigms and at different levels: behavioural, data flow and structural. To allow portability across different foundries, a behavioural description is used. However, a design described behaviourally does not ensure the efficiency of the implementation. HDLs were mainly used in the development of ASICs. To transform HDLs file into netlists format, tools have been developed at an early point of time. This transformation is technically known as synthesis and is part of any design flow as already illustrated in Figure 3.8. Once a boolean code is derived from an HDL description, it needs to be partitioned into an FPGA CLB. This process is more difficult than mapping onto an ASIC library **[29]**.

3.4.2 - Implementation:

It requires translating the already generated Electronic Data Interchange File (EDIF) netlist into an internal format, mapping the captured design onto FPGA chip, placing the gates in the FPGA's CLBs, routing the wires using the PSMs

and finally, generating a bit stream file to configure the FPGA. The whole implementation's process consists of the sequence of the following steps:

- *Translate:* The EDIF netlist is converted into an internal netlist format and different design-rule verifications are performed.
- *Map:* During this process, various optimisations of the logic circuit are carried out either to increase the speed of the circuit and/or decrease the number of gates used.
- *Place and Route:* The gates within the netlist file are assigned to specific CLBs and their interconnections are routed through PSMs and other FPGA's routing resources.
- *Configure:* A bitstream file is generated, and ultimately will be downloaded onto the FPGA chip in order to configure it.

3.4.3 - Programming:

Known also as the configuration process, it consists of downloading the bitsteam file onto the FPGA structure. Finally, the programmed FPGA may be used with a host computer to verify the on chip functionality.

3.5 - EDK: Embedded Development Kit

Embedded systems design is a hot application field which merges logic design and processor-based hardware development in a single or few chips solution. In recent years, embedded applications have emerged at a fast rate and used in every field. Various technologies have been used in the development of embedded systems; microcontroller, DSP processor, ASIC, and now FPGA **[30]**.

Embedded systems development involves hardware and software development as they co-exist in such systems. The software is unique to a system and varies in complexity from system to system and application to application. Field Programmable Gate Arrays (FPGA) from Xilinx started as glue logic usage stitching functions together. Through generations of development, more and more functional blocks were added, increasing applications area envelope. Xilinx FPGA families carry hard core and soft core processors **[11]**. The development kits provide hardware/software tools which

facilitate implementation of functions, which are not yet realized, by embedding them as soft intellectual property in the FPGA fabric. Thus a specific ASIC chip need not be developed for every application, reducing time to market and development cost.

The number of applications combining hardware and software continues to grow. The design of embedded systems often involves design around strict performance, area and power constraints. Often software-heavy implementations are not satisfactory, thus requiring investigation of a mixed hardware software implementation. A mixed hardware-software system is based upon an efficient partitioning of tasks to be implemented in each domain.

The reconfigurability and increasing density of FPGA circuits have made the use of such devices popular for digital system design and prototyping. With the increasing density of FPGAs, they are able to implement larger applications on chip. Earlier efforts in hardware-software co-design have utilized FPGAs as co-processors for acceleration of parts of the application which do not meet the required specifications. The advances in silicon technology have resulted in platform FPGAs, which integrate high density configurable logic, embedded processors, interconnect, dedicated multipliers and block RAMs on a single chip. A single FPGA device can thus be used to target a wide range of applications. Soft processor cores for FPGAs enable customization of processor cores as an application specific embedded processor. The cores can also be encapsulated as intellectual property (IP) and can be easily reused several times on multiple FPGA devices.

FPGA have grown from simple glue logic elements to complex devices able to implement several complex hardware applications. Modern state-of-the-art FPGA devices like Xilinx Virtex FPGAs additionally support a partial dynamic run-time reconfiguration which reveals new aspects for the designer who wants to develop future applications demanding adaptive and flexible hardware. A new approach to create systems that are able to manage configuration is run-time systems. These systems use the flexibility of an

FPGA by changing the configuration partially. Only the necessary functions are configured in the chip's memory. A function can be replaced by another function while other parts stay operative.

The development of partial reconfigurable application on the Xilinx platform is done according to the "modular design flow" provided by Xilinx. The modular design flow provides a guide on how to generate the constraints required by the placement and configuration tools for the generation of full and partial bitstreams representing modules to be downloaded at run-time to fully or partially reconfigure the device. With the increasing interest in software-like hardware design language like C, the need for incorporating this language in the modular design flow process is high.

In our work, we will show how Xilinx FPGAs can be used in embedded applications. The strengths of Xilinx FPGAs and the supporting development tools will be exploited. Design procedures and results will be provided to show the ease of design and development of a complete system using Xilinx Embedded Development Kit (EDK) software.

3.5.1 - System-on-Chip: Overview

System-On-Chip (SoC) on a Field Programmable Gate Array (FPGA) board is an interesting platform that is starting to appear more and more throughout the embedded market. One of the reasons is that it places the processor, memory network connection on one single chip, resulting in a smaller and less power demanding system. Embedded products are therefore a useful area with its often limited space. The SoC solution has existed a long time on Application Specific Integrated Circuit (ASIC) boards but is rather new on FPGA boards. As the FPGA boards have become bigger and faster they are now able to handle a soft processor which is an Intellectual Property (IP) core implemented using logical primitives. A hard processor is faster but less configurable, a soft processor makes it easier to develop and evaluate a solution. A SoC typically consists of a 32-bit core processor and a lot of functions that a designer can choose from.

Advances in semi-conductor industry, specifically on silicon devices, allow to designers the integration of all components present in a computer or

other electronic system into a single chip, this concept has been termed System-on-Chip (SoC). The most common application for SoC is in the embedded systems area. To facilitate the design of SoC systems, the systems are built using existing components that have well-defined contents and interfaces.

The target system used is a SoC board from Xilinx. It consists of many different peripherals such as memory controllers, General Purpose I/O (GPIO) and bus interfaces making it a fitting system in different areas such as: image processing. Its key features are:

- Virtex-II XC2V1000-The chip where all logic connecting the different peripherals and MicroBlazes are placed.
- Memory: Has support for several different memory types, such as: DDR, SRAM.
- Connectors: RS-232 serial port, JTAG, and GPIO consisting of buttons and LEDs.

3.5.2 - The EDK:

The implementation of the Microblaze required the use of Xilinx's Embedded Development Kit (EDK). The EDK is a development environment that provides application designers with the tools necessary to build embedded soft cores processor systems. The steps within the EDK that are necessary to build the embedded processor system include: hardware platform creation, software platform creation, and software application creation. The hardware platform is defined by the Microprocessor Hardware Specification (MHS) file **[11]**. The MHS file defines our system architecture, memory modules, and embedded processors. It also defines the systems connectivity as well as the configurable options and the address map for each memory module in our system. The EDK includes Xilinx Platform Studio (XPS) which is an integrated development environment (IDE). The EDK also provides correct compilers for the target platform. XPS gives a wide selection of settings to modify the kernel as needed. It allows the user to for example initiate interrupt handlers, add thread support, and specify clock resolution. All modifiable settings can be seen in Xilinx OS and libraries documents.

3.5.3 - Platform FPGA's and MicroBlaze

FPGA manufactures, such as Xilinx, have begun introducing FPGAs with architectures capable of providing complete on-chip solutions. These platform FPGA architectures contain clock managers, arithmetic units, embedded memories, processors, high speed I/O … etc. The Virtex-II FPGA in addition to containing the traditional elements that are characteristic of previous Platform FPGA generations contains the MicroBlaze Soft Core Processor Block **[11]**.

The Microblaze (MB) is an embedded soft processor system designed by Xilinx. This processor is an intellectual property (IP) core that is implemented using the logic primitives of the FPGA. The architecture is shown in Figure 3.9.

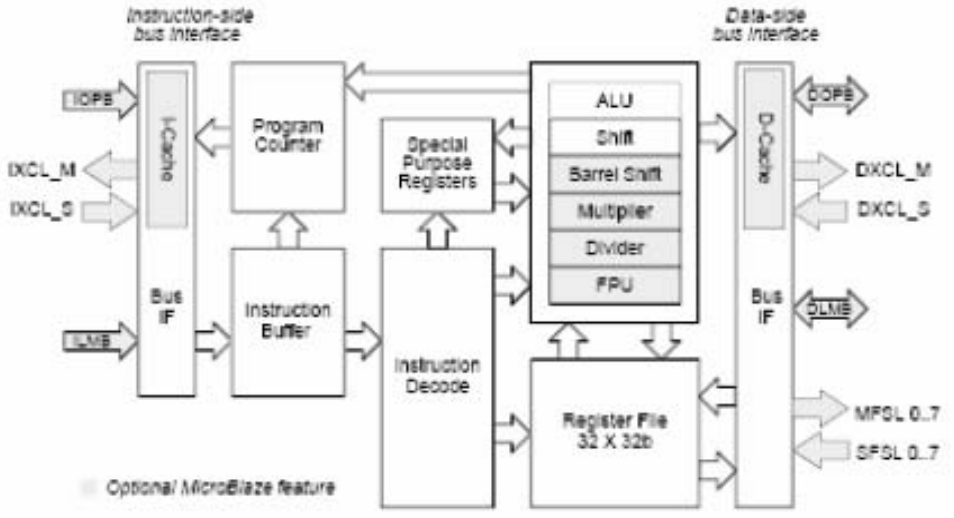


Figure 3.9 : MicroBlaze Core Block Diagram **[11]**

The Microblaze is a processor system that introduces an integrated single precision, IEEE-754 compatible Floating Point Unit (FPU). Its core is a 3-stage pipeline, 32-bit RISC Harvard architecture soft processor core with 32 general purpose registers, Arithmetic Logic Unit (ALU), and an instruction set optimized for embedded applications. It supports both on-chip block RAM and external memory. The MicroBlaze has the On-Chip Peripheral Bus (OPB) to interface all the different devices and custom logic. Depending on the configured options, the MicroBlaze will use between 900 and 2600 Look-Up

Tables (LUTs) and the number of processors on a single FPGA is only limited by the size of the FPGA.

3.5.4 - SRAM

A traditional processor requires external memory for storage of application data, and hence a Static Random Access Memory (SRAM) is required in combination with the processor core implemented on the FPGA. In our design only one SRAM device was chosen to keep the initial design and interface simple. The SRAM can be tested by writing data (image) to a chosen address location on the SRAM and reading back from that location. The MicroBlaze supports word, halfword, and byte accesses to data memory. It also supports both on-chip block RAM and external memory.

3.5.5 - Floating Point Unit (FPU)

The Microblaze introduces an integrated single precision, IEEE-754 compatible Floating Point Unit (FPU). Applications created using floating-point operations in software have a higher execution time. FPU has a low latency resulting very useful in these cases.

3.5.6 - Channel communications

Channels provide a link between parallel branches. One parallel branch outputs data onto the channel and the other branch reads data from the channel. Channels also provide synchronization between parallel branches because the data transfer can only complete when both parties are ready for it. If the transmitter is not ready for the communication then the receiver must wait for it to become ready and vice versa. In Figure 3.10, the channel is shown transferring data from the left branch to the right branch. If the left branch reaches point ***a*** before the right branch reaches point ***b***, the left branch waits at point ***a*** until the right branch reaches point ***b***.

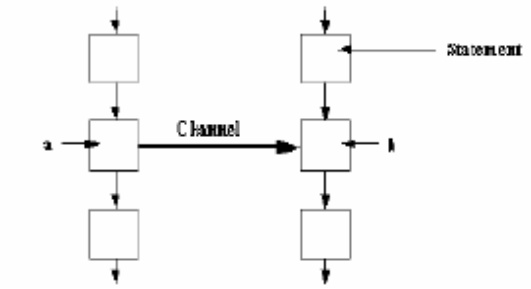


Figure 3.10: Channel Communication

3.5.7 - External Communication

Communication between the hardware and the outside world is performed using interfaces. These may be specified as input or output, and, as with assignment, a write-to or a read-from an interface will take one clock cycle. The language allows the designer to target particular hardware, assign input and output pins, specify the timing of signals, and generally control the low level hardware interfacing details. Macros are available to help target particular devices.

3.5.8 - Some Restrictions When Using C and FPGA

One problem with C is that it is not designed as a HDL, rather a high-level language with a hardware output. Since C targets hardware, it imposes some programming restrictions when compared to a traditional C compiler. These need to be taken into consideration. Some of these restrictions particularly affect the implementation of algorithms. Firstly, there is no stack available, so recursive functions cannot be directly supported by the language and can not be implemented without some modifications. Secondly, the size of memory that can be implemented using standard logic cells on an FPGA is limited, because implementing memory is an inefficient use of FPGA resources. However, some FPGAs have internal RAM that can be used by C. A limitation of using RAM or ROM is that it cannot be accessed more than once per clock cycle.

3.6 - FPGA Implementations of DWTs:

Most of wavelet designs and implementations reported in the literature up to date are dedicated to fully custom hardware (VLSI-oriented). A full range of such systems was briefly reviewed. The implementations presented are somewhat unclear due to several factors:

- The information is commercial or proprietary.
- Word-length constraints have been omitted to enable an improvement in area over previous designs to be reported.

- The levels of decomposition are very low, thus showing a better hardware utilisation over a design having more levels of decomposition.

The choice of a particular wavelet can affect the hardware performance by requiring a different number of MACs.

3.7 - Summary:

The emergence of high density, high performance FPGAs provides an attractive alternative for implementing complex signal and image processing applications. Despite their low-level model, these flexible devices have several advantages over traditional DSPs. They provide signal and image processing designers with the flexibility of software while keeping intact the hardware performances. In this chapter, a brief overview on FPGAs based configurable engine has been introduced. The benefits of this new computing concept over traditional computing have also been highlighted.

Even though FPGA devices are supplied by different manufacturers, they share several features. In general, FPGA devices consist of a 2-D array of logic cells, each of which contains enough resources to implement a single or more bit operations. These logic cells are connected through the usage of different routing resources. The communication with the external "world" is ensured by versatile I/O elements surrounding the device. The architecture of Xilinx Virtex II FPGA (used through this thesis) have been described in this chapter.

CHAPTER 4

ES SALAM RESEARCH REACTOR UTILISATION FOR NUCLEAR IMAGING

4.1 - INTRODUCTION :

Research Reactors are the nuclear reactors which are not used to produce electricity. The first objectif of a RR is to rather bring knowledge than producing heat or electricity. We generally regard a research reactor as a significant precondition to the development of a nuclear energy program. The neutron flux within the reactor is sufficiently high to have nuclear advantages like the production of radioisotopes for medical and industrial uses.

Es salam research reactor has diverged for the first time in 1992. Es Salam is a research reactor primarily intended to provide neutron beams to accomplish needs for fundamental and applied research, for production of radioelements, for material tests and also for training in nuclear engineering.

Es Salam research reactor, conceived to meet these needs, is a multi purpose reactor, of tank type and a nominal output power of 15 mW (thermal) with a flux of $2x10^{14}$ $n/cm^2/s$. It is moderated and cooled with heavy water. It offers new experimental possibilities in the field of neutron spectrometry, production, and materials study. The neutron beams will be used by chemists, biologists, metallurgists, and physicists to study the structure and dynamics of condensed matter, theoretical physics, neutron physics, nuclear reactions and physics of elementary particles.

4.2 DESCRIPTION:

The core contained in an aluminium tank is composed of 72 fuel assemblies comprising fuel elements whose fissile material is consisted of dioxide uranium UO_2 enriched to 3%. The reactivity control is carried out by means of absorbing cadmium bars moving vertically in their tube guides. The fuel is surrounded by heavy water as a cooler and moderator with a layer of graphite as a reflector. A biological shielding is

ensured by light water contained in side, lower and higher tanks and a heavy concrete wall is build outside.

4.2.1 - Horizontal channels:

The Es Salam RR, primarily intended for supplying neutron beams, is equipped with six horizontal channels and a thermal column, one of which is used for neutron imaging.

4.2.2 - Vertical channels:

Twenty three vertical channels, being inside the reactor core, can be used for the irradiation of fuel assemblies, radioelements production and analysis of samples by neutron activation. Twenty vertical channels, in graphite reflector, can be also used. Two vertical channels, being in water tanks can be used as biological channels.

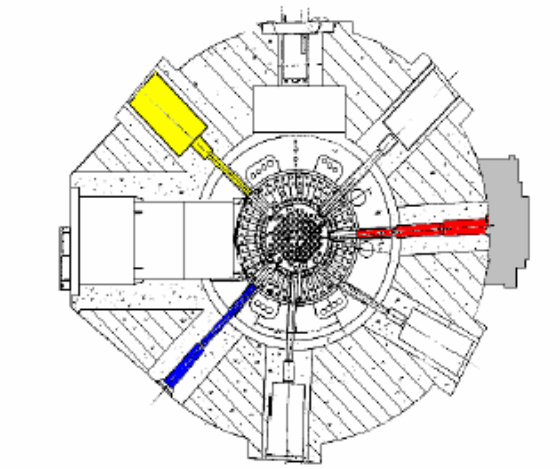


Figure 4.1: Horizontal channels

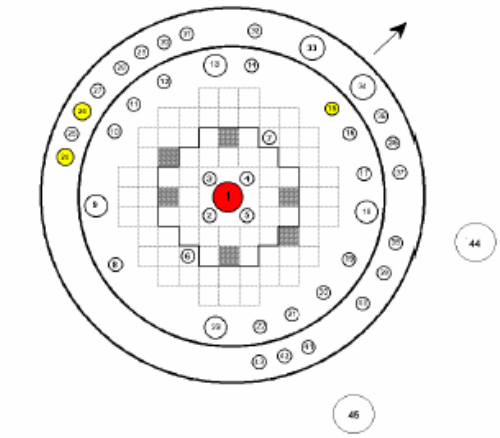


Figure 4.2: Vertical channels

4.3 - NEUTRON RADIOGRAPHY:

Neutron beam photography is a means of non destructive testing which, in its principle, is similar to an examination by X or gamma rays, but the produced image is different. The specificity of Neutron beam photography resides in neutron flux disturbance by the body to be examined, which is basically different from that of photon flux. The difference is at the level of interaction between the radiation used and the crossed matter. This technique produces complementary results and often contrary to those of radiography, in particular in certain applications such as: the control of hydrogenated bodies under metalic enclosures, level of filling of containers, location of joints, wiring control. It is an imaging method, by transparency, of materials or of heterogeneous industrial devices, using the contrast of neutron diffusion or absorption of various elements. A beam of thermal neutrons irradiates a sample or a piece, the transmitted attenuated beam, is recorded in the form of 2D image. This image reproduces the space variations of sample attenuation, therefore makes it possible to visualize the localization of absorbing or diffusing elements. In general, there is a converter, and the 2D image is recorded on a photographic film.

The opaque matter scanned with x or gamma rays are far from delivering all its secrets: certain bodies let easily pass these rays, others absorb them completely. Neutron beam photography makes it possible to very clearly see the structure of these materials; indeed the slow neutrons are easily absorbed by hydrogenated bodies and rather easily cross those of strong densities whereas x-rays are unusable in these two particular cases. When the beam of neutrons crosses the object to be studied, it undergoes attenuations which vary from a point to another according to the thickness and the various matters which are made of.

Applications:

- In electrical engineering: it is used for checking the electric insulation integrity in components, checking of the quality of contact in assemblies of the metal or composite parts.
- In lubricants industry: allows the visualization of deterioration of oil films, seizing zones, presence of liquid or organic deposits in assemblies or piping.

- In Metallurgy: for the control of welding between two different metals of close densities (Zirconium - stainless for example), the control of welding and alloy assembly with strong density (platinum iridium for example), the low density alloy control (borated aluminium for example), alloy homogeneity with strong neutron contrast (boron intensified steels, steels gadolinium).
- In archaeology: is used for the qualitative and quantitative examinations of archaeological parts and art objects.
- In Biology: Qualitative examination of biological species, such as for example the study of circulation metabolism and water transfer in flowers for increase their shelf life after gathering. Examination of bones of certain prehistorically mammals is also a field where neutron beam photography can provide much information.

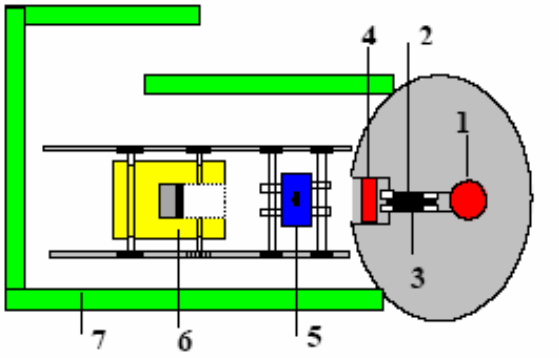


Figure 4.3: Neutron radiography installation

1- Reactor Core. 2- Horizontal channel. 3- Flux Collimator. 4- Channel door. 5- Exposition table. 6- Beam catcher. 7- Biological protection.

4.4 - NEUTRON Tomography:

A neutron Tomography facility is actually under construction around Es-Salam research reactor. The developed state-of-the-art image processing in this memory will be used around this facility.

Due to rapid development of VLSI technology as well as development in the field of digital image processing, new methods and techniques are nowadays available which have the potential to improve the image quality and investigation capabilities in neutron

tomography as well as radio isotopic imaging. Image processing and analyzing methods to be used around the future Es-Salam neutron tomography facility. Neutron transmission experiments were performed around the Neutron radiography. Our approach for image compression helps a lot for 2D image storage and transmission. It is based on projection images collection, encoding/decoding, TCP/IP transmission and therefore conversion to suitable format for 3D reconstruction. Different projection images quality enhancement were applied at the raw level of images. These operations have the advantage to reduce background and artifacts contributions in the final 3D image reconstruction result.

Neutron Tomography is dedicated to resolve the problem of spatial structures visualisation along the direction of the beam that is impossible with Neutron Radiography. In contrast, computed tomography enables to look at cross-sectional images or “slices” of an object without physically cutting it. The principle of computed tomography (CT) is based on 3D image reconstruction of an object through its neutron transmission data obtained for different projection angles. By recording the transmission images (projections) of an object from many different angles, it is possible to mathematically reconstruct the distribution of attenuation coefficient within an object. By assembling several slices, a 3-D volumetric image of the inner structure of the object can be built up **[31]**.

Neutron Tomography Principle (NT):

Neutron Tomography (NT) is a technique of a 3D measurement of neutron attenuation coefficient, using reconstruction calculation from projection images at different angles (Fig. 1) **[32]**. NT, similar to X-ray CT, uses the transmitted radiation modulation of an object to describe macroscopic object details **[33]**. The only difference resides in the fact that in X-rays Tomography the source and the detector turn around the object while for neutrons the beam is fixed and the object turns. X-rays are replaced by neutrons as the penetrating radiation in a through-transmission examination. Since the absorption characteristics of matter for X-rays and neutrons differ drastically, the two techniques can be considered as complementary.

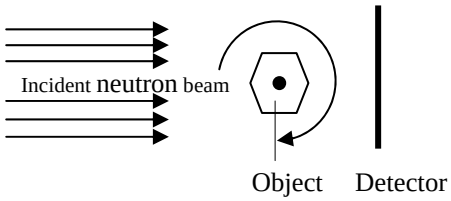


Figure 4.4: Principle of neutron tomography

A typical tomography system is composed of a neutron source, a neutron collimation system, a scintillator screen, an object turntable, a mirror, a cooled CCD camera and a computer support **[34]** , see figure 4.5.

For every projection, the transmitted neutron intensity reaches the scintillator screen and the generated light is reflected by the mirror and recorded by the cooled CCD camera. To take the necessary number of projections, the object is generally turned with a constant step from 0° to 180°.

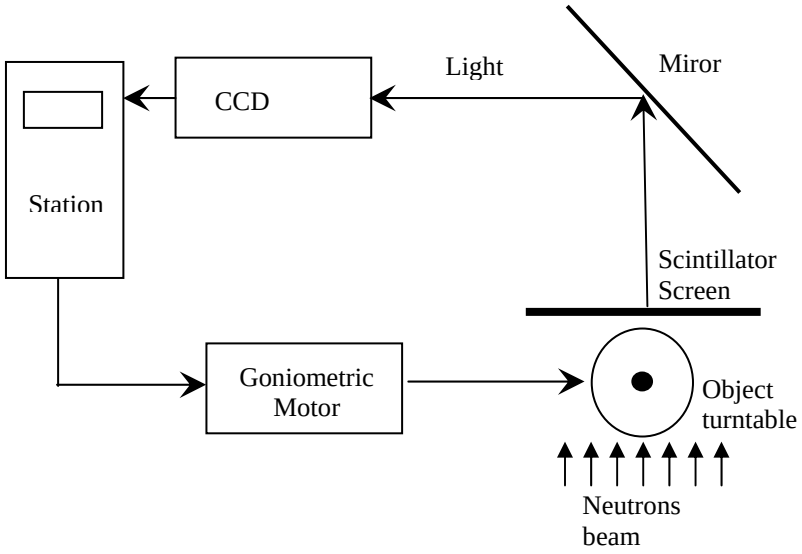


Figure 4.5: Neutron Tomography System

4.5 - RADIOISOTOPE:

Radioisotopes are atoms that contain an unstable combination of neutrons and protons.The combination can occur naturally, as in radium-226, or by artificially altering the atoms. In some cases, a nuclear reactor is used, in others, a cyclotron. Atoms

containing this unstable combination regain stability by shedding radioactive energy, hence the term radioisotope. The process of shedding the excess radioactive energy is called decay. The radioactive decay process of each type of radioisotope is unique and is measured with a time period called a half-life.

A radiopharmaceutical is a molecule that consists of a radioisotope tracer attached to a pharmaceutical. After entering the body, the radio-labelled pharmaceutical will accumulate in a specific organ or tumour tissue. The radioisotope attached to the targeting pharmaceutical will undergo decay and produce specific amounts of radiation that can be used to diagnose or treat human diseases and injuries. The amount of radiopharmaceutical administered is carefully selected to ensure each patient's safety. Radioisotopes are an essential part of radiopharmaceuticals.

A sealed radioactive source is an encapsulated quantity of a radioisotope used to provide a beam of ionising radiation. Industrial sources usually contain radioisotopes that emit gamma or X-rays.

Ionising radiation is radiation which can knock electrons out of atoms, either by direct interaction with the atoms or by other methods. Alpha and beta particles, neutrons, X-rays and gamma rays are examples of ionising radiation.

Applications of isotopes:

They are widely used in medicine, in industry and for scientific research, and new applications for their use are constantly being developed. In many cases, radioisotopes have no substitute and in most of their applications they are more effective and cheaper than alternative techniques or processes. Some radioisotopes used in nuclear medicine have very short half-lives, which mean they decay quickly; others with longer half-lives take more time to decay, which makes them suitable for therapeutic purposes.

⟹ *In industry:*

Industry uses radioisotopes in a variety of ways to improve productivity and gain information that cannot be obtained in any other way. Radioisotopes are commonly used in industrial radiography, which uses a gamma source to conduct stress testing or check the integrity of welds – a common example is to test aeroplane jet engine turbines for

structural integrity. Radioisotopes are also used by industry for gauging (to measure levels of liquid inside containers) or to measure the thickness of materials or in mineral analysis. Radioisotopes with high gamma ray levels are also used in the radiation sterilisation of medical supplies and food packaging.

Radioisotopes are also widely used in scientific research, and are employed in a range of applications, from tracing the flow of contaminants in biological systems, to determining metabolic processes in small animals.

⟹ *In nuclear medicine:*

Nuclear medicine uses small amounts of radiation to provide information about a person's body and the functioning of specific organs, ongoing biological processes, or the disease state of a specific illness. In most cases, the information is used by physicians to make an accurate diagnosis of the patient's illness. In certain cases radiation can be used to treat diseased organs or tumours.

Diagnostic radiopharmaceuticals can be used to examine blood flow to the brain, to assess functioning of the liver, lungs, heart or kidneys, to assess bone damage, and to confirm other diagnostic procedures. They are used in sports medicine to diagnose stress fractures, which are not generally visible in X-rays.

4.6 - NUCLEAR IMAGING:

Nuclear imaging is a technique that uses radioisotopes that emit gamma rays from within the body. To make a radiopharmaceutical, a radioisotope is attached to a pharmaceutical that is taken up by a specific organ or specific diseased tissues. The radiopharmaceutical is given orally, injected or inhaled, and is detected by a gamma camera which is used to create a computer-enhanced image that can be viewed by the physician.

There is a significant difference between nuclear imaging **[21]** and other medical imaging systems such as CT (computerised tomography), MRI (magnetic resonance imaging) or X-rays. Nuclear imaging measures the function (by measuring blood flow, distribution or accumulation of the radioisotope) of a part of the body and does not provide highly resolved anatomical images of body structures. PET scans are frequently combined with CT scans, with the PET scan providing functional information (where the

radioisotope has accumulated) and the CT scan refining the location. The primary advantage of PET imaging is that it can provide the examining physician with quantified data about the radiopharmaceutical distribution in the absorbing tissue or organ. The information obtained by nuclear imaging tells an experienced physician much about how a given part of a person's body is functioning. By using nuclear imaging to obtain a bone scan for example, physicians can detect the presence of secondary cancer 'spread' up to two years ahead of a standard X-ray. It highlights the almost microscopic remodelling attempts of the skeleton as it fights the invading cancer cells. The main difference between nuclear imaging and other imaging systems is that, in nuclear imaging, the source of the emitted radiation is within the body. Nuclear imaging shows the position and concentration of the radioisotope. If very little of the radioisotope has been taken up a 'cold spot' will show on the screen indicating, perhaps, that blood is not getting through. A 'hot spot' on the other hand may indicate excess radioactivity uptake in the tissue or organ that may be due to a diseased state, such as an infection or cancer. Both bone and soft tissue can be imaged successfully with this system.

4.6.1 - GAMMA CAMERA:

An imaging apparatus used to visualize the distribution of radionuclides within the body. The majority of gamma cameras in clinical use operate on the principle devised originally by H.O. Anger at the Donner Laboratory in Berkeley in 1956. Radiation emanating from the patient is detected by a single, large, circular NaI (Tl) scintillation crystal. An array of (37, 61 or 91) photomultiplier tubes detects the light quanta emitted by the crystal and converts their energies into electrical pulses. Associated electronic circuitry determines the x, y coordinates of each scintillation event, the outputs of all the tubes being summed to provide a z pulse, the amplitude of which corresponds to the total energy of the scintillation event. The distribution of radioactivity is usually displayed on an oscilloscope. For quantitative analysis, the gamma camera is connected to a dedicated microcomputer, radioactive distributions being displayed either on a monochrome TC monitor as a gray scale range of tomes or on a color TC monitor as a color scale.

Gamma radiography camera works in a similar way to the X-ray camera used to scan luggage at airports. A small pellet of radioactive material in a sealed titanium capsule is positioned on one side of the object being screened. A sheet of photographic film is placed on the other side. The gamma rays are beamed from the radioactive source and pass through the object to create an image on the film. Just as X-rays show a break in a bone, gamma rays show flaws in metal castings or welded joints. The technique allows critical components to be inspected for internal defects without damaging the component or making it radioactive.

Gamma radiography has several advantages over an X-ray camera. X-ray cameras are large and cumbersome and require a high voltage electrical charge source. Radioactive sources, on the other hand, are small and do not require power. This means the radioisotopes can be transported easily to remote areas and used where there is no power. They can also be located inside equipment to produce photographs of internal joints or components without the need for dismantling.

4.6.2 - POSITRON EMISSION TOMOGRAPHY (PET):

PET, or Positron Emission Tomography, is one of the most sophisticated medical imaging technologies available today. The radioactive components of radiopharmaceuticals used in PET procedures are usually made in cyclotrons. PET cameras can be located in some of our hospitals.

PET cameras are extremely sensitive. They can be used to detect very early signs of disease and to map how organs such as the brain and heart are functioning. Most radioisotopes used with PET have short half-lives.

PET is based on the detection, by means of opposed detectors and coincidence counting techniques, of the two 511kev photons which are simultaneously emitted in almost opposite directions by a positron emitting radionuclide.

4.6.3 - SINGLE PHOTON EMISSION COMPUTED TOMOGRAPHY (SPECT):

Is the most commonly used form of tomographic imaging, SPECT cameras are usually used with radiopharmaceuticals that have longer half-lives than those used with PET.

Single Photon Emission Computerized Tomography is a technique that uses a computer for tomographic reconstruction (in a variety of planes, transaxial, coronal, sagital) of the distribution of a single photon gamma emitting radionuclide detected by a rotating gamma camera (the widely used Tc-99m, which emits single 140 kev gamma photons, is an example). Its essential goal is enhancement of the image detectability and the extraction of quantitative data from a true three dimensional distribution of structure (or radioactivity) in space.

4.6.4 - Digital Radiological Images:

A digital radiologic image is a digital image acquired by a certain radiologic procedure. It is a two-dimensional MxN array of non-negative integers f(x,y), where x and y are the coordinates of anatomical structures in the image. The image segment represented by the coordinates (x,y) is called a picture element, or a pixel, and f(x,y) is its functional value or gray level. The radiologic procedure can be X-rays, ultrasound, computerized tomography, nuclear magnetic resonance, or another digital modality. Depending on the digitization procedure or the radiologic procedure, the gray level can range from 0 to 255, 0 to 511, 0 to 1023, 0 to 2047, and 0 to 4095. These gray levels represent some physical or chemical properties of the object structure. As an example, in an image obtained by digitizing an X-ray film, the gray level value of a pixel denotes the optical density of the square area of that film. For X-ray computerized tomography (XCT), the gray level value represents the relative linear attenuation coefficient of the tissue. For magnetic resonance imaging (MRI), it corresponds to the magnetic resonance signal response of the tissue. The size of an image and the number of images taken in one patient examination vary with modalities. In contrast with most other types of biomedical images, a radiologic image is monochrome, i.e., there is no need to do color compression. All these practical constraints motivate the search for efficient compression for economic storage and fast transmission of radiologic images.

4.6.5 - DIAGNOSIS:

Diagnostic techniques in nuclear medicine use radioactive tracers which emit gamma rays from within the body. These tracers are generally short-lived isotopes linked to

chemical compounds which permit specific physiological processes to be scrutinised. They can be given by injection, inhalation or orally. The first type are where single photons are detected by a gamma camera which can view organs from many different angles. The camera builds up an image from the points from which radiation is emitted; this image is enhanced by a computer and viewed by a physician on a monitor for indications of abnormal conditions.

4.7 - REACTOR AND CYCLOTRON-PRODUCED RADIOISOTOPES:

The great majority of medically useful radioisotopes are made in a nuclear research reactor, however cyclotrons also produce radioisotopes that complement those manufactured in nuclear research reactors. The nucleus of an atom contains two types of particles – neutrons and protons. Stable atoms have a stable ratio of neutrons and protons in the nucleus, while unstable atoms have an unstable ratio. Scientists make radioactive atoms by adding either extra neutrons or extra protons. Atoms with extra neutrons in the nucleus are neutron-rich and are produced in a nuclear reactor. Atoms with extra protons in the nucleus are neutron-deficient and are produced in a particle accelerator such as a cyclotron. Neutron-rich and neutron-deficient radioisotopes decay by different means and thus have different properties and different uses.

The end use of the radioisotope determines the radioactive properties required, and hence whether a nuclear reactor or a cyclotron is used to produce the radioisotope. Over 80% of the radioisotopes actually used in medical procedures worldwide come from reactors. The most commonly used radioisotope, molybdenum-99 (which decays into technetium 99-m) can only be produced economically in a nuclear research reactor. Also, the emerging generation of therapeutic isotopes can only be produced in such a reactor.

CHAPTER 5

DESIGN AND IMPLEMENTATION OF DWT BASED ENCODER / DECODER

5.1 - Introduction

As mentioned previously, the implementation of a forward DWT (or inverse DWT) stage is achieved through the usage of a FIR based filter banks. In such structures, the decimation (or interpolation) by a factor of 2 results in the even indexed samples being multiplied by the even indexed coefficients and the odd indexed samples being multiplied by the odd indexed coefficients. To avoid a waste of data when carrying out the multiplication operations, a pre-processing operation is required. Generally, this consists of splitting the incoming data into even and odd indexed parts. Therefore, only the necessary operations are carried out and the decimation is performed outside the structure. This is made possible by the use of the so-called polyphase decomposition scheme **[17]**.

Another approach that preserves the resources during the filtering process is also possible. In this state the decimation task is performed within the structure. In this case, the pre-processing operation of splitting the incoming data into even and odd parts is not needed anymore. This is made possible by folding the even and odd parts of each filter of the filter bank.

In this chapter, an implementation for both forward and inverse DWT is presented. It is suitable for implementing any wavelet filter belonging either to the orthonormal basis or biorthogonal basis. Furthermore, the presented structure is highly scalable even when more than one level is used. Also are presented in this chapter the implementation performances.

5.2 - Finite Impulse Response Filters:

Finite Impulse Response digital filters (FIR) are the basis in digital signal/image processing based systems **[35]**. In general, they are used to perform different tasks, ranging from high-pass and low-pass filtering, convolution and decimation to more complex purposes like DCT, FFT and wavelet implementations **[36]**. To implement wavelet filter banks, FIRs combined with decimators by a factor of two are generally used. These FIRs present different structures of which the three basic types are reviewed in this chapter.

Basically, the time domain representation of a digital filter is given by the following convolution product:

$$y_n = h_n * x_n = \sum_{i=-\infty}^{+\infty} h_i . x_{n-i} \tag{5.1}$$

where x_i, y_i, and h_i, are the input data variable, the filter output and the filter's coefficients, respectively. The combination of equation 5.1 with a decimator by a factor of two, which is represented by $\downarrow 2$, leads to a "decimated" version of equation 5.1:

$$y_n = h_n * x_{2n} = \sum_{i=-\infty}^{+\infty} h_i . x_{2n-i} \tag{5.2}$$

In this case, only one output is computed for two consecutive input signal samples. On the other hand, the combination of equation 5.1 with an interpolator, generally represented by $\uparrow 2$, leads to the an "interpolated" version of equation 5.1:

$$y_{2n} = h_{2n} * x_n = \sum_{i=-\infty}^{+\infty} h_{2i} . x_{n-i} \tag{5.3}$$

5.2.1 - Canonical Structure

A direct implementation of equation 5.2 can be performed using the structure illustrated in Figure 5.1. To allow the convolution between a filter coefficient and the appropriate input (x_n x_{n-1} x_{n-2} x_{n-3} ...), a line of delay elements, represented by the symbol Δ are added. The structure represented in Figure 5.1 is called the canonical form of a FIR.

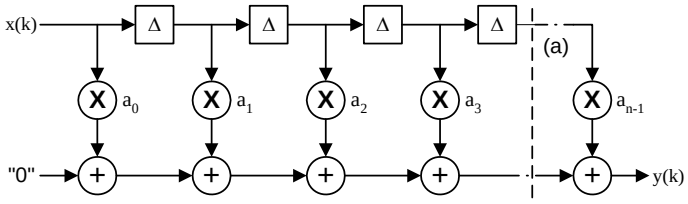


Figure 5.1: Canonical Form

Due to the chain of adders used to accumulate the consecutive convoluted samples, the delay along a multiplier and the delay along n adders span the critical path of this structure. If T_a, T_m represent the delay of an adder and a multiplier,

respectively, the critical delay T_{cr} is equal to $nT_a + T_m$. This structure is suitable for implementing systems with a linear phase property such as biorthogonal wavelets.

5.2.2 - Inverted Structure

An alternative inverted structure of the canonical structure illustrated in Figure 5.1 can be derived by a simple transposition. In this particular case, the delay elements are inserted between two consecutive adders and the coefficient allocation is reversed. The inverted form is illustrated in Figure 5.2.

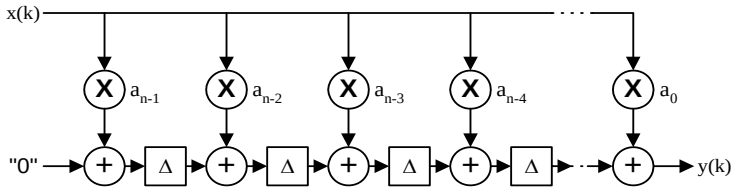


Figure 5.2: Inverted Form

The critical path for the inverted form is spanned by the delay of a multiplier and an adder. Consequently, the critical delay T_{cr} is equal to $T_a + T_m$. This structure is suitable for implementing pipelined designs with a minimum number of registers.

5.2.3 - Pipelined Structure

Another alternative to the canonical and the inverted forms described previously is the pipelined structure as illustrated in Figure 5.3. This structure can be derived either from the canonical structure or from the inverted structure leading to two new different forms. However, since the inverted structure is already pipelined, a cut set line for pipelining purposes is applied vertically between two consecutive sets of adder-multiplier (illustrated in Figure 5.1).

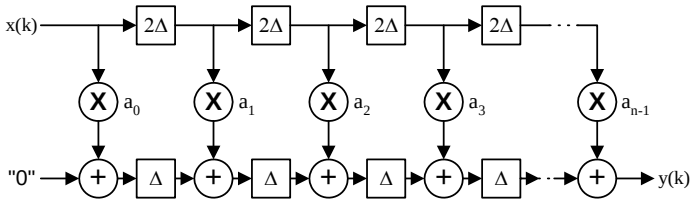


Figure 5.3: Pipelined Form

Since two consecutive adders are “cut” by a delay element, the critical path is reduced to an adder and a multiplier. Therefore, the critical delay T_{cr} is equal to $T_a + T_m$ as in the case of the inverted form shown in section 5.2.2.

5.3 - Polyphase Decomposition Scheme

When a wavelet transform operation is processed in a conventional way, half of the outputs are discarded at the end due to the decimation process. Obviously,

this leads to a waste of resources. To remedy to this problem the polyphase decomposition scheme is used **[17]**. It allows us to carry out the decimation operation prior to the filtering operation thus saving half of the computational data. Thus, an output is computed for every two input samples. This doubles the throughput rate of the filter bank. In the case of the wavelet transform, the polyphase decomposition scheme is illustrated in Figure 5.4. $E_0(z)$ and $E_1(z)$ represent the minimal transfer functions which an even and an odd indexed sample has to come across, respectively.

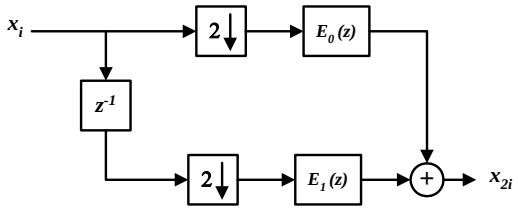


Figure 5.4: Polyphase Decomposition Scheme

5.4 - Architecture for Forward DWTs

As reported in Chapter 2, the Forward Discrete Wavelet Transform (FDWT) of a signal decomposes the original signal by means of a hierarchy of wavelet signals obtained through a multistage signal decomposition process **[8][18]**. These wavelet signals characterise the detail and approximation signals corresponding to different resolution stages. Thus, a signal x is decomposed into an approximation 'a' and a detail 'd' both of length $N/2$ (the low pass and the high pass filters are supposed to be of equal lengths) by means of the following two convolutions:

$$a_n = \sum_{i=0}^{N-1} h_i . x_{2n-i} \tag{5.4}$$

and

$$d_n = \sum_{i=0}^{N-1} g_i . x_{2n-i} \tag{5.5}$$

h_i, g_i, d_i and a_i represent the low pass coefficients, the high pass coefficients, the detail outputs and the approximation outputs, respectively.

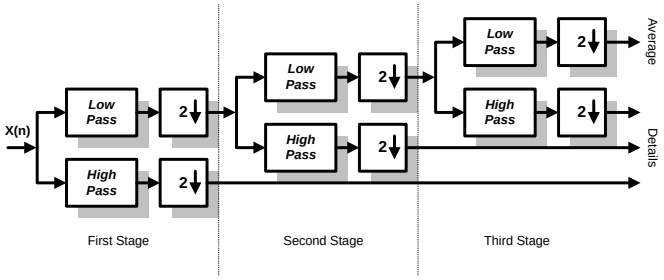


Figure 5.5: A three stage 1-D Wavelet analysis system

5.4.1 - Decimated FIR's:

Initially, due to the fact that the even indexed samples are convoluted with the even indexed coefficients and the odd indexed samples are convoluted with the odd indexed coefficients, a direct implementation of equations 5.4 and 5.5 leads obviously to a waste of data (as stated previously). In fact, a direct implementation of these two equations means simply that half of the filtered output samples have to be discarded at the end of the filtering process. This is due to the presence of a decimating operator, which is represented by the index $(2n-i)$. This problem is solved by the use of the polyphase decomposition scheme introduced in section 5.3. The application of this scheme can be formalised by the following equations:

$$a_n = \sum_{i=0}^{\lfloor N-1/2 \rfloor} h_{2i}.x_{2n-2i} + \sum_{i=0}^{\lfloor N-1/2 \rfloor} h_{2i+1}.x_{2n-2i-1} \tag{5.6}$$

and

$$d_n = \sum_{i=0}^{\lfloor N-1/2 \rfloor} g_{2i}.x_{2n-2i} + \sum_{i=0}^{\lfloor N-1/2 \rfloor} g_{2i+1}.x_{2n-2i-1} \tag{5.7}$$

The $\lfloor x \rfloor$ operator represents the highest integer lower than the value of x.

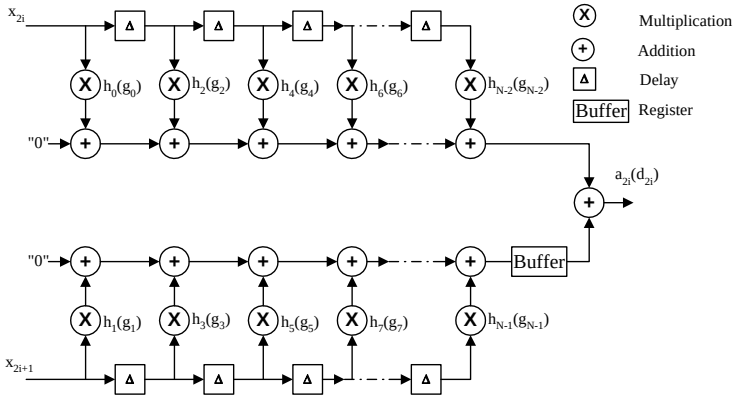


Figure 5.6: Architecture of a decimated FIR filter
(The number of coefficients N is supposed to be even)

The filtering-decimating implementation process as represented by equation 5.5 and equation 5.7 can be achieved simply by splitting the signal samples into even and odd indexed parts. The resulting architecture of such approach is illustrated in Figure 5.6.

To hold the data correctly for the appropriate multiplication time, the signal samples are held using a delay element before being processed by the next multiplier in the chain.

5.5 - Architecture for Inverse DWTs

Unlike in the case of the analysis or the decomposition step described previously, the data in the synthesis step is up-sampled (or interpolated) before being processed by the corresponding high pass and low pass filters.

Furthermore, the outputs of both filters are merged to form a wavelet stage output, which acts as the reconstructed average input to the succeeding stage. The other input (detail) of the wavelet stage comes from the corresponding resolution. Figure 5.7 shows a typical 3-stage 1-D wavelet synthesis structure.

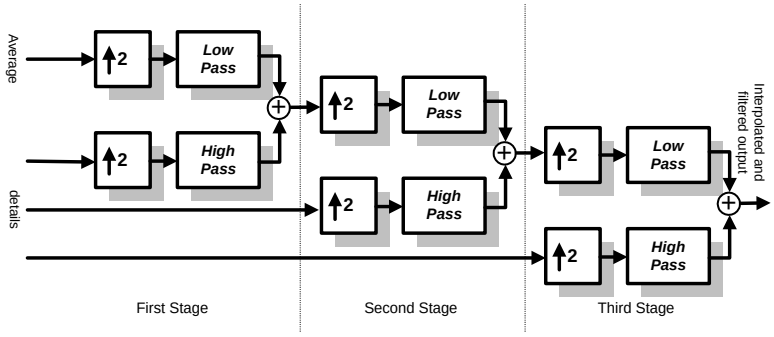


Figure 5.7: A three stage 1-D Wavelet synthesis system

In general, this inverse process requires the introduction of a zero sample between each two consecutive data samples. This can be achieved by making a “duplicate” of each data sample, thus allowing it to be multiplied by two filter coefficients. This procedure can be formalised by the two following equations:

$$x_{2n} = \sum_{i=0}^{\lfloor N/2-1 \rfloor} \tilde{g}_{2i}.d_{n-i} + \sum_{i=0}^{\lfloor N/2-1 \rfloor} \tilde{h}_{2i}.a_{n-i} \tag{5.14}$$

And

$$x_{2n+1} = \sum_{i=0}^{\lfloor N/2-1 \rfloor} \tilde{g}_{2i+1}.d_{n-i} + \sum_{i=0}^{\lfloor N/2-1 \rfloor} \tilde{h}_{2i+1}.a_{n-i} \tag{5.15}$$

5.6 - Generation of Wavelet Filter Banks

To generate a wavelet filter (forward or inverse), a partition based on the parameters that affect different parts of the filter need be defined first. In addition, this partition has to take into consideration the similarities between the different parts of the filters of the two bases. This allows the “reuse” of a particular part during the implementation of a wavelet filter bank **[38]**.

As it has been shown previously, both forward and inverse wavelet filters are partitioned into three parts: a Processing Element (multipliers and full adders), a Terminating Element, and a chain of Delay Elements.

5.7 - Multilevel Processing

It is well known that in DCT-based applications the signal/image of interest needs to be processed once in order to get the corresponding frequency information. However, when transforming a signal/image using the wavelet transform, the same processing is repeated several times using a “similar” structure at each level (multiresolution). The main problem to deal with here is the “growth” of the internal word length from a level to another.

5.8 - Extending the Model to 2-D Wavelet Transforms

The 2-D wavelet transformation does not differ in essence from the 1-D wavelet transformation. In both cases it consists of convoluting a sliding vector to the set of data samples followed by a decimation process. Moreover a 1-D wavelet transformation can be used to carry out 2-D one. In image processing applications to transform an image from the temporal domain to the joint scale-frequency domain, a 2-D wavelet transformation is required. This operation can be carried out using two different methodologies. The first one is the so-called the standard decomposition. It consists by first transforming all the rows of the image and then transforming the columns of the resulting image. The second one is the non-standard decomposition and consists of transforming the rows and the columns of the image alternatively. Figure 5.8 illustrates a decomposition stage used to compute the 2-D DWT.

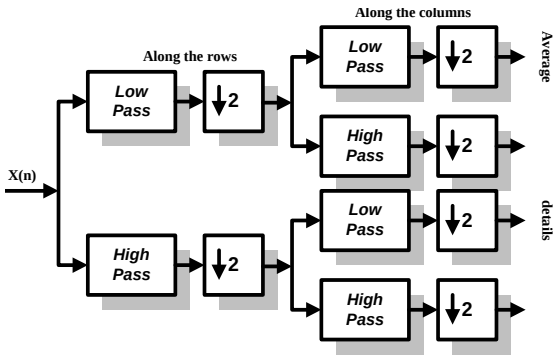


Figure 5.8: 2-D DWT decomposition stage

5.10 - Software implementation:

Matlab, a popular mathematical modeling software, is used for the software implementation of the discrete wavelet transform and all the encoding stages. In the memory files format, pixels are stored in the unsigned char type, providing a maximum of 256 gray scale levels or 8-bit data per pixel. The Matlab programming environment provides all the necessary functions and tools needed to achieve this work. But, unfortunately, we have not the toolbox necessary for converting Matlab programming codes to VHDL language codes due to the absence of an updated licence.

The wavelet transform routine implements a modified version of the biorthogonal (2-2) Cohen-Daubechies-Feauveau wavelet and the Daubechies Wavelet for test images. No special boundary treatment is performed on the images while evaluating the DWT.

5.10.1 - Biorthogonal Cohen-Daubechies–Feuveau wavelet:

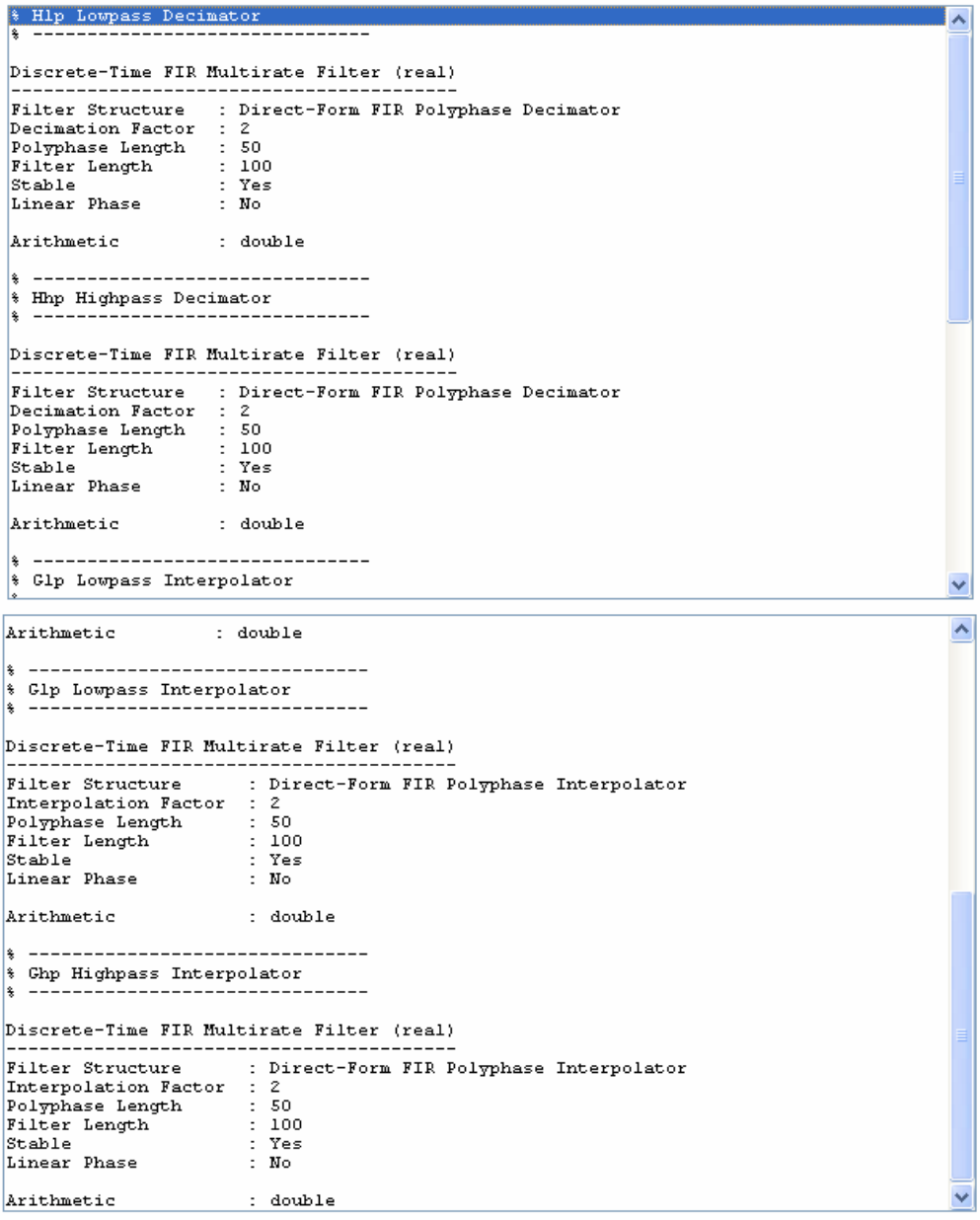
The test images used are both 256X256 and 512X512 pixels. The Biorthogonal Cohen-Daubechies–Feuvear wavelet acts as a series of additions and subtractions to generate the average and detail coefficients. Figure 5.15 shows both the original 2-D projection image from CCD neutron tomography system and the wavelet transform on this image at level -3 decomposition.

5.10.2 - Daubechies wavelet:

The same test images are used. The Daubechies wavelet acts as a series of high and low pass filter to generate the average and detail coefficients.

A Matlab program has been implemented **[40]** to perform a 3-level multiresolution wavelet transform followed by quantization and a Huffman encoding. A second part composed of Huffman decoding, dequantization and inverse wavelet transform. It is the functionality shown in this program that the FPGA designed

system in this thesis attempts to imitate. When executed, this program produces images at any selected stage of the encoder, as shown in figures 5.17 and 5.18. The first is the original image before any processing has been performed. The third one is the result after decoding at the receiver. This last image should not match the original image since the wavelet transform followed by quantization is not reversible (and also due perhaps to some slight floating point arithmetic rounding errors).



```
% Hlp Lowpass Decimator
% ------------------------------

Discrete-Time FIR Multirate Filter (real)
-----------------------------------------
Filter Structure   : Direct-Form FIR Polyphase Decimator
Decimation Factor  : 2
Polyphase Length   : 50
Filter Length      : 100
Stable             : Yes
Linear Phase       : No

Arithmetic         : double

% ------------------------------
% Hhp Highpass Decimator
% ------------------------------

Discrete-Time FIR Multirate Filter (real)
-----------------------------------------
Filter Structure   : Direct-Form FIR Polyphase Decimator
Decimation Factor  : 2
Polyphase Length   : 50
Filter Length      : 100
Stable             : Yes
Linear Phase       : No

Arithmetic         : double

% ------------------------------
% Glp Lowpass Interpolator
%
```

```
Arithmetic         : double

% ------------------------------
% Glp Lowpass Interpolator
% ------------------------------

Discrete-Time FIR Multirate Filter (real)
-----------------------------------------
Filter Structure      : Direct-Form FIR Polyphase Interpolator
Interpolation Factor  : 2
Polyphase Length      : 50
Filter Length         : 100
Stable                : Yes
Linear Phase          : No

Arithmetic            : double

% ------------------------------
% Ghp Highpass Interpolator
% ------------------------------

Discrete-Time FIR Multirate Filter (real)
-----------------------------------------
Filter Structure      : Direct-Form FIR Polyphase Interpolator
Interpolation Factor  : 2
Polyphase Length      : 50
Filter Length         : 100
Stable                : Yes
Linear Phase          : No

Arithmetic            : double
```

Figure 5.9: The QMF FIR filters used to implement the DWT

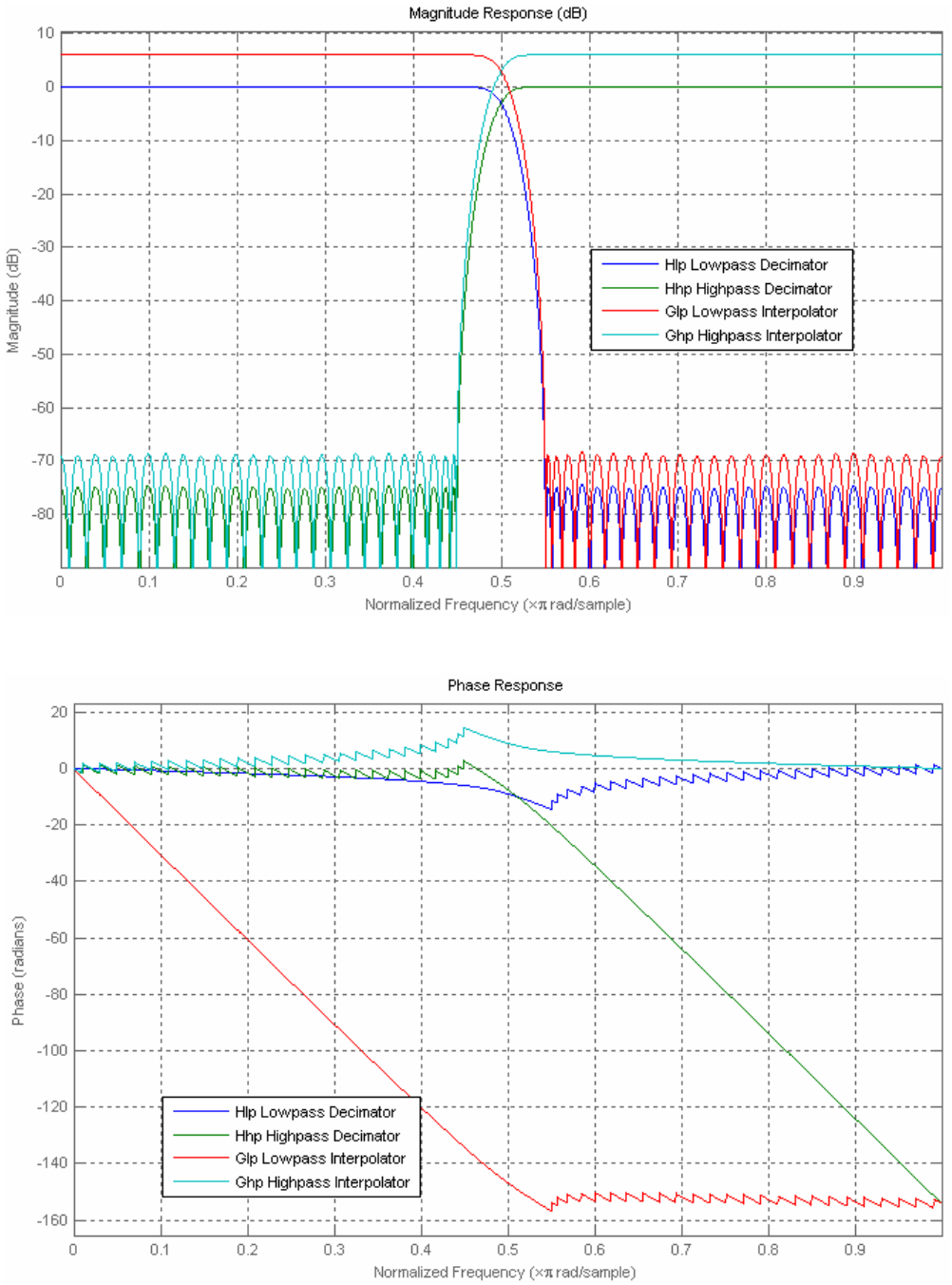


Figure 5.10: The QMF FIR filters used: Magnitude and Phase response

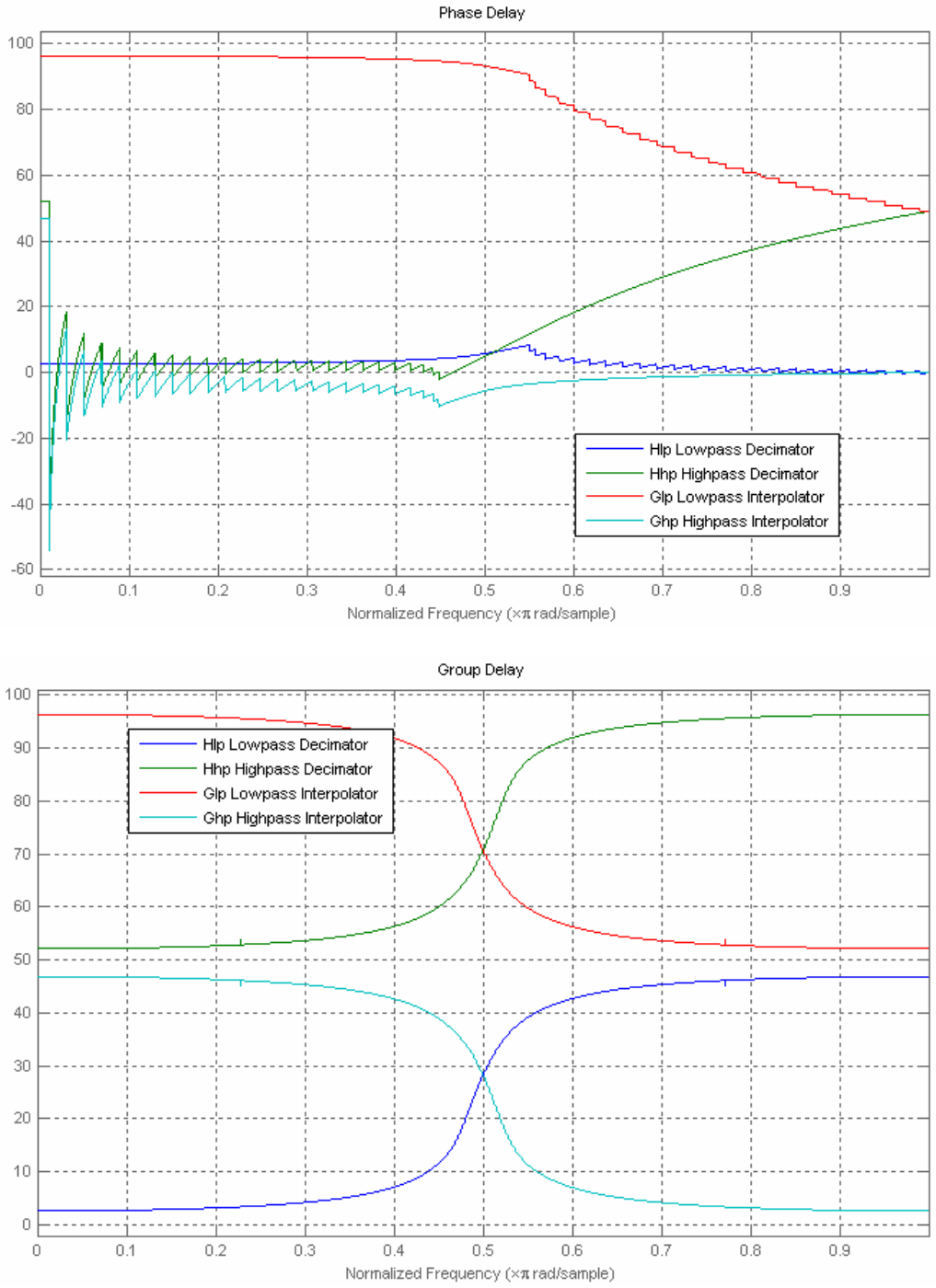


Figure 5.11: The QMF FIR filters used: Phase and Group delays

5.11 - Design partitioning

The whole computation is partitioned into many stages. The first stage reads raw images from CCD camera of stored files and computes its discrete wavelet transform coefficients. The other stages operate on this result to complete the rest of the processing: dynamic quantization, zero thresholding, run length encoding for

zeroes, and entropy encoding on the coefficients, then the encoded image is transmitted through a TCP/IP network. In the same time it is stored in a local database system.

Starting with a raw image, the coefficient values are calculated and put in variables accessed by the second stage. Compression involves truncating wavelet coefficients below a threshold. An experiment conducted on an image, shows that most of the coefficients have small magnitudes. More than 90% of the wavelet coefficients have less than 5% of the maximum value. This means that most of the signal energy is in the high-valued coefficients, which are few. Thus the small valued coefficients can be truncated or zeroed and then be used to reconstruct the signal.

The coefficients from different sub-bands are quantized separately. The dynamic range of the coefficients for each sub-band (computed in first stage) is divided into 16 quantization levels. The coefficients are quantized into one of the 16 possible levels. The maximum and minimum value of the coefficients for each sub-band is also needed while decoding the image.

The quantizer stage of the image compression process works by performing a thresholding operation on the transformed image. Fundamentally, it performs a mapping from a continuously-parameterized set to a discrete set. All values within a fixed range are set to a single pre-determined value. Values outside of a pre-determined maximum and minimum boundary are set to the respective boundary value. The price of this gain in compression ratio is the loss of image richness as neighboring color values are merged into a single intensity. This generic thesholding process of quantization is shown in Figure 5.9. An example of the selective quantization approach is shown in Figure 5.10. Where sub-bands are either quantized by 2, 4, 8, or 16. What this means in practice is that an integer division operation (i.e. bit shift) is performed, and the least-significant bits of the image pixels are simply discarded. This provides the desired level of quantization.

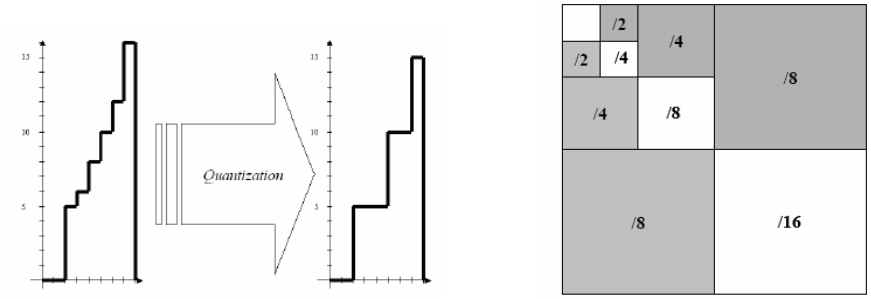


Figure 5.12: Thresholding,quantization and Variable quantization with sub-bands

Regions with abrupt changes will have larger wavelet coefficients while regions of little or no change would have smaller coefficients. Coefficients of small magnitude can be neglected without considerable distortion to the image. The error introduced is proportional to the magnitude of the coefficient being neglected. After the zero thresholding, a large number of coefficients are truncated to zero. Long sequences of zeroes can be effectively compressed by run length encoding, which replaces each individual occurrence of a zero with a count. To decode a run length encoded stream, this count has to be distinguishable from other characters of the input data set. The other valid characters are the output from the quantizer. Whenever RLE detects a zero, it starts counting the sequence of continuous zeroes.

The encoder is the only stage of the process that actually achieves compression. It does this by removing redundancy present in the image. Two common algorithms that are used in conjunction with wavelet image compression are Stack-Run and Huffman encoders. Stack-run encoders use sequential scanning within subbands to reduce the number of bits required to represent the image. This algorithm is effective in reducing long streams of zeros, which the wavelet transform and quantization stages commonly produce. Following the application of Stack-Run Encoding, traditional image compression processes often employ Huffman coding, table 5.2, also known as entropy encoding. This process, through the systematic use of binary trees, attempts compress the bit stream by assigning shorter bit patterns for the most common data elements, and longer bit patterns for the less frequent data elements. This involves variable length encoding of the input data.

Original source			Source reduction			
Sym.	Prob.	Code	1	2	3	4
a_2	0.4	1	0.4 1	0.4 1	0.4 1	0.6 0
a_6	0.3	00	0.3 00	0.3 00	0.3 00	0.4 1
a_1	0.1	011	0.1 011	0.2 010	0.3 01	
a_4	0.1	0100	0.1 0100	0.1 011		
a_3	0.06	01010	0.1 0101			
a_5	0.04	01011				

Table 5.13: Example of Huffman coding

5.12 - Matlab interface:

The main program is small in volume, it consists of loading a GUI interface and calling the raw image on which we apply the necessary functions for encoding and decoding:

4) ***Wavelet transform (bi-orthogonal Cohen-Daubchies-Feauveau).***
5) ***Quantize: quantizes the elements of matrix.***
6) ***Entropy function: computes a first order estimate of the entropy of a matrix.***
7) ***Huffman encoding: builds a variable-length Huffman code for a matrix.***
8) ***Huffman decoding: decodes a Huffman encoded matrix.***
9) ***Dequantize: dequantizes the elements of matrix.***
10) ***Wavelet inverse transform (bi-orthogonal Cohen-Daubchies-Feauveau).***

Following is some additional functions for calculating some compression metrics such as:

1) ***Compression ratio: computes the ratio of the bytes in two images/variables.***
2) ***Root mean square error: computes and displays the error between two matrices.***
3) ***Peak to signal to noise ratio.***

5.13 Metrics:

The metrics used to evaluate compression quality are:

MSE: Mean Square Error.

$$\text{MSE} = \frac{1}{512\text{x}512} \sum_{x=1}^{x=512} \sum_{y=1}^{y=512} [p(x,y) - p'(x,y)]^2$$

Where p(x,y) and p'(x,y) are the two pixel level gray values.

RMSE: Root Mean Square Error.

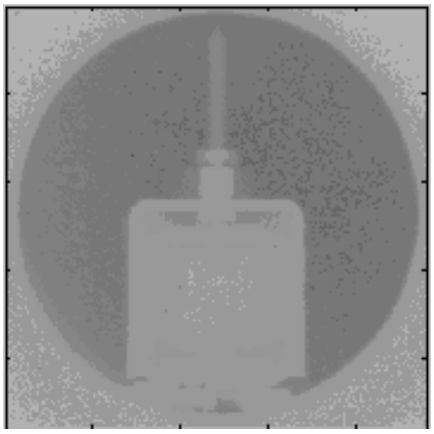
$$\text{RMSE} = \sqrt{\text{MSE}}$$

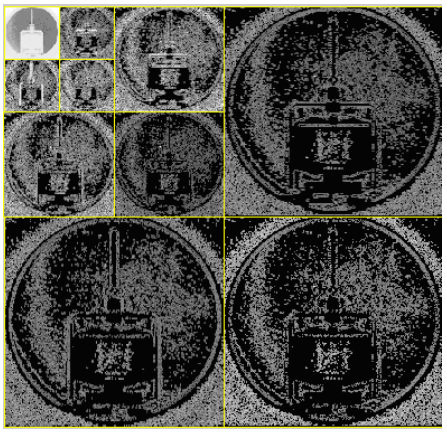
PSNR: Peak Signal to Noise Ratio.

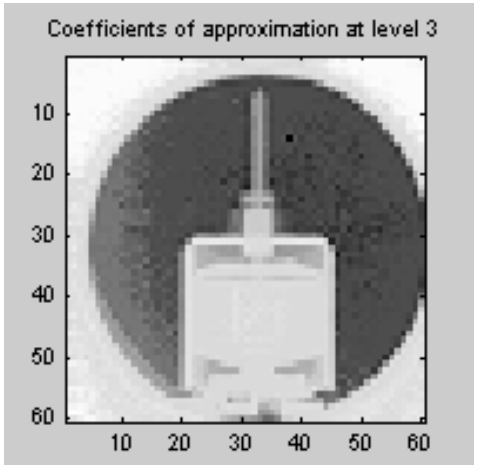
$$PSNR = 20.\log_{10}\left(\frac{255}{RMSE}\right)$$

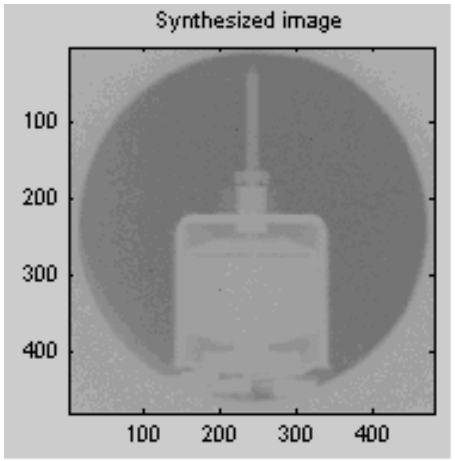
For one dimensional signal, these metrics become:

$$PSNR = 20.\log_{10}\left(\frac{255}{\sqrt{\frac{1}{N}\sum_{N}(x_i - \hat{x}i)^2}}\right)$$









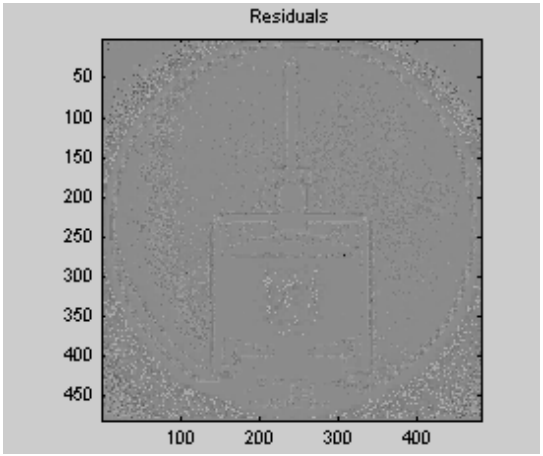


Figure 5.13: Original, 3-level DWT, approximation and decoded images with details

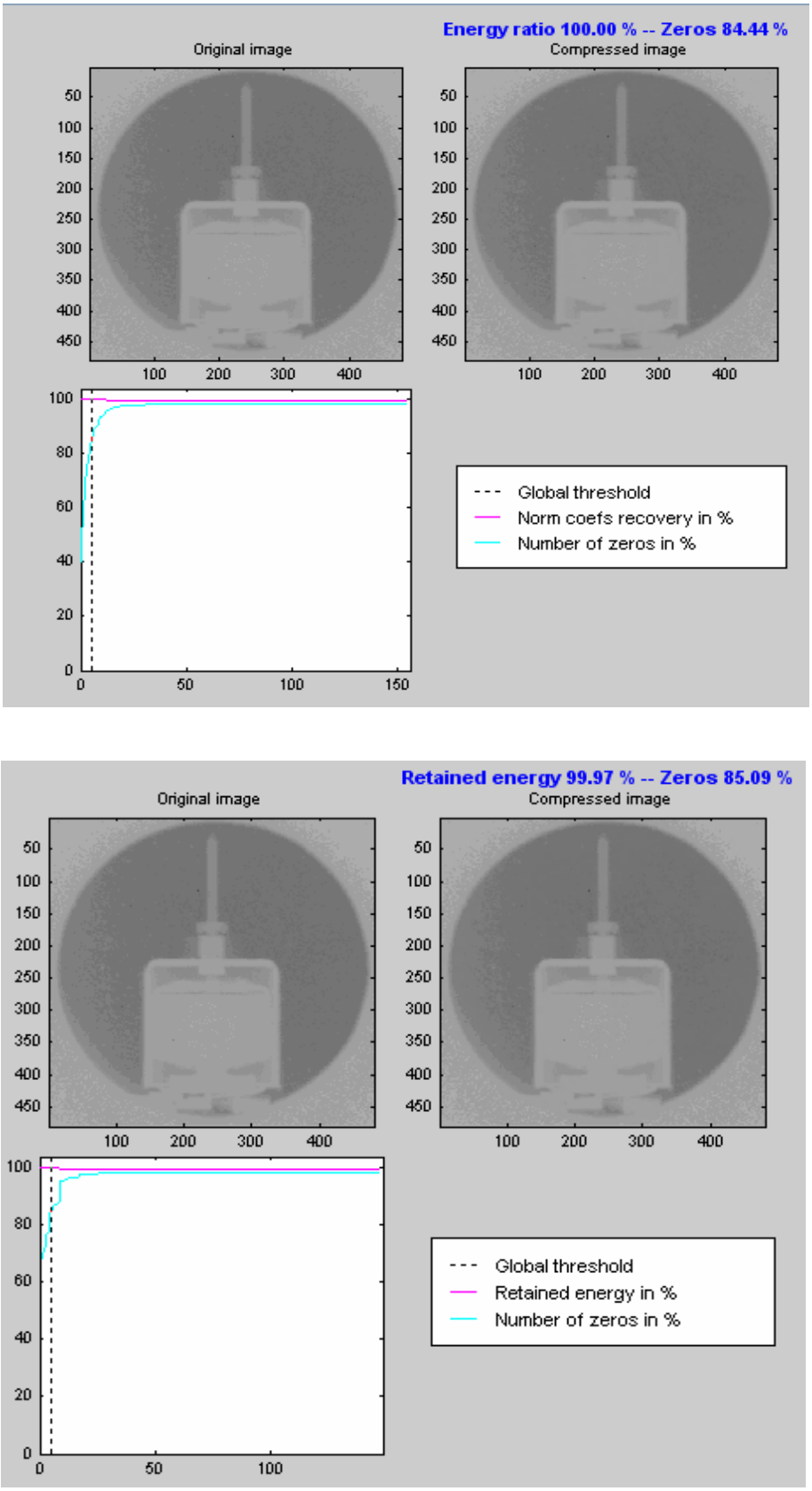


Figure 5.14: The retained energy with varying quantization threshold

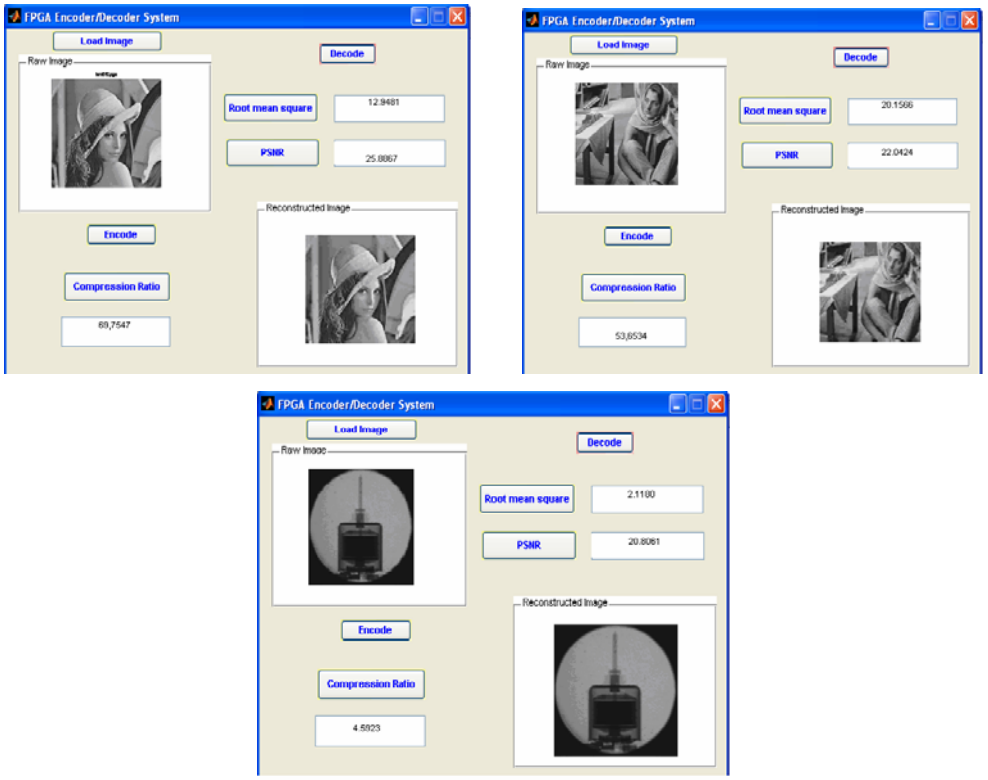


Figure 5.15: Encoder/Decoder execution for lena, barbara and one projection NR

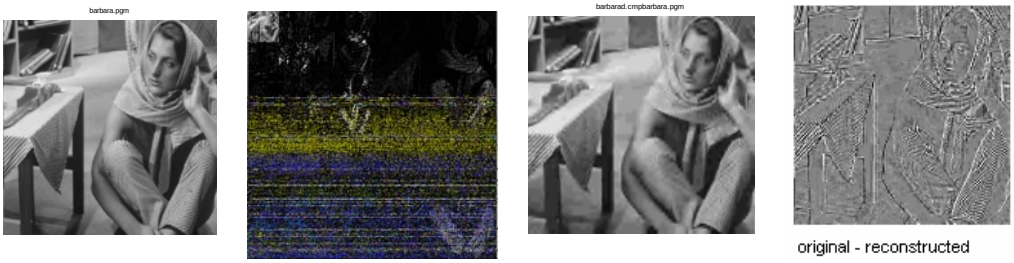


Figure 5.16 : Barbara image and its level 3 DWT, reconstructed image, and the difference between original and reconstructed one.

Figure 5.17 : Lena image and its level 3 DWT, reconstructed image, and the difference between original and reconstructed one.

Images	barbara	Lena
Size	256X256	512X512
Decomposition level	3	3
MSE	406.2898	167.6531
RMSE	20.1566	12.9481
PSNR	22.0424	25.8867
Compression ratio	53,6534	69,7547

Table 5.3: comparison between two different size images

Image type	Compression Ratio (CR)	Root Mean Square Error
Cameraman image (standard): 256/256	8	4.2361
Neutron Image: 152x165	4.5923	2.1180
Isotope image: 242x242	10.6791	8.4721
C-Tomography 3D image: 256x256	2.6667	4.1727

Table 5.4: comparison between different type images

5.15 - Summary

A software implementation of the whole Encoder/Decoder system has been designed, realized and tested and presented in this chapter. A VHDL implementation of the Discrete Wavelet Transform based on MALLAT's Algorithm can be driven from this implementation in order to be loaded on Xilinx Virtex-II FPGA, after simulation on ModelSim6.0 and synthesize using ISE 8.1. Timing report, performance report in addition to device utilization reports can also be verified with the floor plan mapping.

Our vision in implementing the Encoder – Transmission - Decoder system was to use the functionality offered by MATLAB simulink in wavelet image transform. But faced to the absence of a recent license for HDL generation Toolbox, we moved to another FPGA implementation possibility following the steps below:

1 – Use MATLAB implementation for results comparison (Compression Ratio, Root Mean Square Error and Peak to Signal to Noise Ratio).

2 – Implement the system using standard C high level language and then programming on FPGA using the Embedded Development Kit (EDK) based on a system on chip architecture.

3 - Exploit the MicroBlaze processor property of the Virtex-II FPGA by building a system platform for our application implemented on the PC and running on the FPGA. This will be the subject of the next chapter.

CHAPTER 6

ENCODER / DECODER IMPLEMENTATION USING THE XILINX EDK (EMBEDDED DEVELOPMENT KIT)

6.1 - Introduction

Over the last few years there have been several attempts at translating algorithmic oriented programming languages directly into hardware descriptions. Like hardware description language, C allows the designer to focus more on the specification of an algorithm rather than adopting a structural approach to coding. Our goal is to implement all image Encoder/Decoder algorithms: DWT, quantizer, run length, and Huffman encoder on FPGA using C language and compare results against the performance of software implementation on a general purpose computer (PC) using Matlab.

Implementing image processing algorithms on reconfigurable hardware minimizes the time-to-market cost, enables rapid prototyping of complex algorithms and simplifies debugging and verification.

6.2 - Programming FPGA with C language:

The algorithms were developed using the EDK development environment and was implemented on Xilinx Vertex-II FPGA based board. The reason for selecting C language is its standard syntax. The standard C is used for the development of a host program which can be verified and its results can be compared to the hardware version.

A host side program on the PC was written (also using standard C) for FPGA configuration and communication with FPGA board. The board device driver routines provided by the vendor are employed to accomplish this task.

The software system on the PC, Pentium 4 /3.3 GHz with 512M RAM, consists of:

- Operating System: Window XP professional edition.
- Matlab 7.0 .
- Compiler Tools : Dev-C++ for Standard C / C++ programming.
- Hardware Synthesis Tools : Xilinx XST for Xilinx FPGAs, Xilinx EDK for soft-core synthesis for Xilinx FPGAs.
- Xilinx Embedded Development Kit (EDK): Xilinx Platform Studio (XPS) tools.
- Software Development Kit (SDK), Eclipse-based IDE.
- MicroBlaze.

6.2.1 - The objected system design scheme

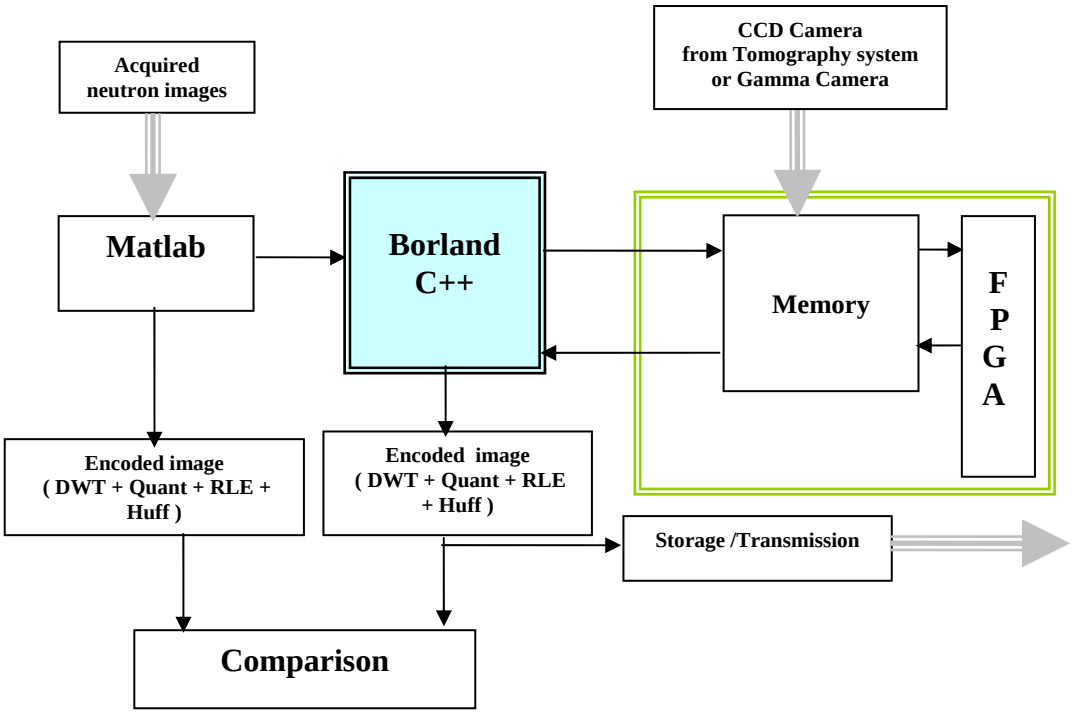


Figure 6.1: The Overall design scheme for the Encoder / Decoder

The two C-programs for encoding and decoding are build using the same organization, each one is composed of a main() calling some necessary functions:

Encoding functions are:

1) Read_image: opens the specified image file, calculates the size of image and makes a copy of the image as integers in an array of data.
2) wavelet transform: using the biorthogonal cohen daubchies feauveau wavelet.
3) Quantization: using a scale function of 16 levels.
4) Run length encoding: calculates the number of zeros by incrementing a counter.
5) Variable length encoding: or Huffman codes using the fixed encoding tree of JPEG2000.
6) Write_image: writing the Huffman codes in a file, this file will be used by the decoding program.

Decoding functions:

1) read_ compressed_file:
2) Huffman decoding:
3) Run length encoding:
4) Dequantize

5) Inverse wavelet transform.
6) Write image.

An optional compression factor, varying from 0 to 255, is entered to see how aggressively the image should be compressed.

6.2.2 - From software to hardware

In this and the following part we will pull together some of the concepts and create a hardware/software implementation of our Encoder/Decoder system. We will show how a C code that was not originally written with a hardware implementation in mind can be iteratively ported, compiled, simulated, and refined to create a reasonably efficient hardware implementation.

The original source code was written in standard C language and was not optimized for any specific processor target. The algorithm was designed to encode/decode 8-bit pixels of 512X512 pixels image.

The change made to the C algorithm in support of hardware compilation is how to load the input image data to the memory and how to read the compressed output data by the host computer.

After simulating its functionality using standard desktop tools, we are ready to implement the application on a mixed FPGA/processor target. We chose a Xilinx MicroBlaze-based FPGA target for this test, selecting the Virtex-II MicroBlaze Development Kit (EDK) as our reference system. This kit includes a hardware reference board populated with a Virtex II FPGA and various peripheral interfaces, as well as all development tools needed to compile and synthesize hardware and software applications (consisting of HDL source files for hardware and C source files for software) to the FPGA target. When combined with the C compiler, this kit provided us with everything needed to compile and execute our application. **[41]**.

6.3 - MicroBlaze Bench Creation

In this part we will explain the process of creating and testing a MicroBlaze system design. We will go through the process, step-by-step, of creating a platform using the Xilinx Embedded Development Kit (EDK) tools **[41]**. The following steps are used in our work:

- Starting XPS (Xilinx Platform Studio: available with the FPGA kit).
- Using the Base System Builder Wizard, create a basic Microblaze plat forme.
- Create IP Peripherals, depending on what we need.
- Implementing the Designed modules and adjusting parameters.
- Defining the Software Design
- Downloading the Design on the board.
- Debugging the Design for functionality.

• Performing Behavioral Simulation of the Embedded System (we do also structural or timing simulation).

• Implementing the application: Image Encoder and Decoder: adding sources.

We can define the embedded system hardware with the Microprocessor Hardware Specification (MHS) and Microprocessor Peripheral Description (MPD) files. These files describe the following:

• Embedded processor: the soft core MicroBlaze processor.

• Peripherals and associated address spaces.

• Buses.

• Overall connectivity of the system.

The MHS file is a readable text file that is an input to the Platform Generator. Conceptually, the MHS file is a textual schematic of the embedded system. Each system peripheral has a corresponding MPD file. The MPD file is the symbol of the embedded system peripheral to the MHS schematic of the embedded system. The MPD file contains all of the available ports and hardware parameters for a peripheral.

Create or Import IP Peripheral: One of the key advantages of building an embedded system in an FGPA is the ability to include customer IP and interface that IP to the processor.

Design Modification using Platform Studio: Once a design has been created with the Base System Builder, it can be modified within the System Assembly view.

6. 4 - Building the User Application

Now that the hardware has been completely specified in the MHS file, we can run the Platform Generator. Platform Generator elaborates the MHS file into a hardware system consisting of NGC files that represent the processor system. Then the Xilinx ISE tools will be called to implement the design for the target board.

XPS provides the ability for the user to create multiple software projects. These projects can include source files, header files, and linker scripts. Unique software projects allow the designer to specify the following options for each software project:

• Specify compiler options.

• Specify which projects to compile.

• Specify which projects to download.

• Build entire project.

6.5 - Downloading and running the Application

Now that the hardware and software designs are completed, the device can be configured. Following these steps to download and configure the FGPA:

• Connecting the host computer to the target board, including connecting the Parallel-JTAG cable and the serial cable.

• Starting a hyperterminal session connecting our development machine to the serial port connected to the RS232 peripheral on the main FPGA development board with the following settings:

* Com1: This is dependant on the com port your serial cable is connected to.
* Bits per second: 57600
* Data bits: 8
* Parity: none
* Stop bits: 1
* Flow control: none

• Connecting the board power and turn on.

• Select the Download function from the Tools menu in Platform Studio.

• In the process window, double click on Update Bitstream with Processor Data.

• In the process window, double click on Configure Device (iMPACT) under Generate Programming File.

• With iMPACT configure the FPGA, and when the device is configured, we can debug the software application directly via the JTAG connections.

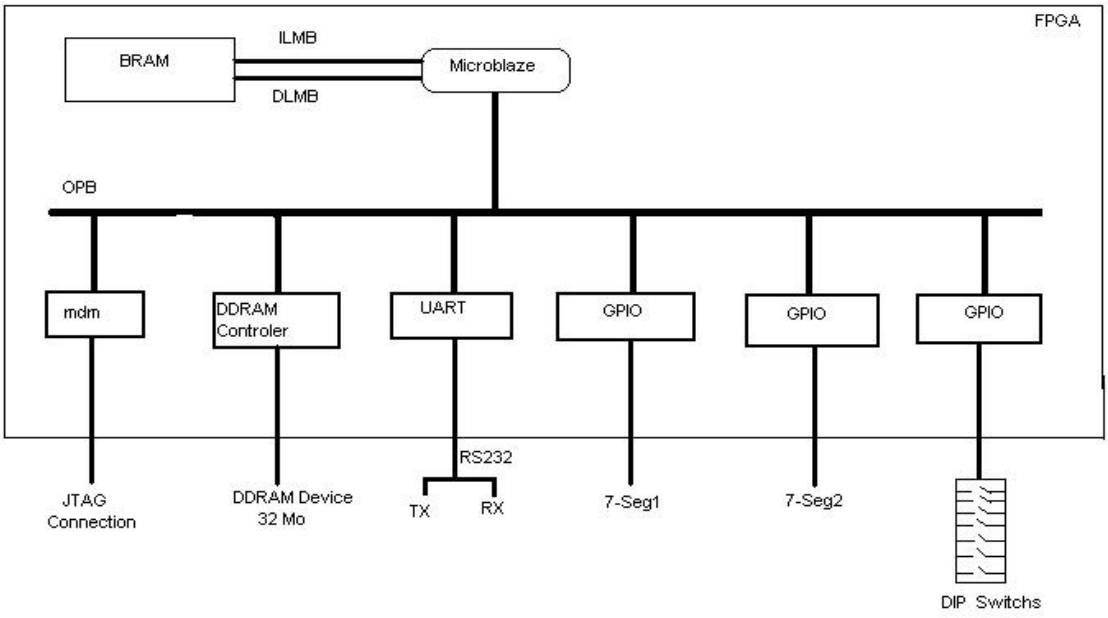


Figure 6.2: The created hardware plate form simplified scheme

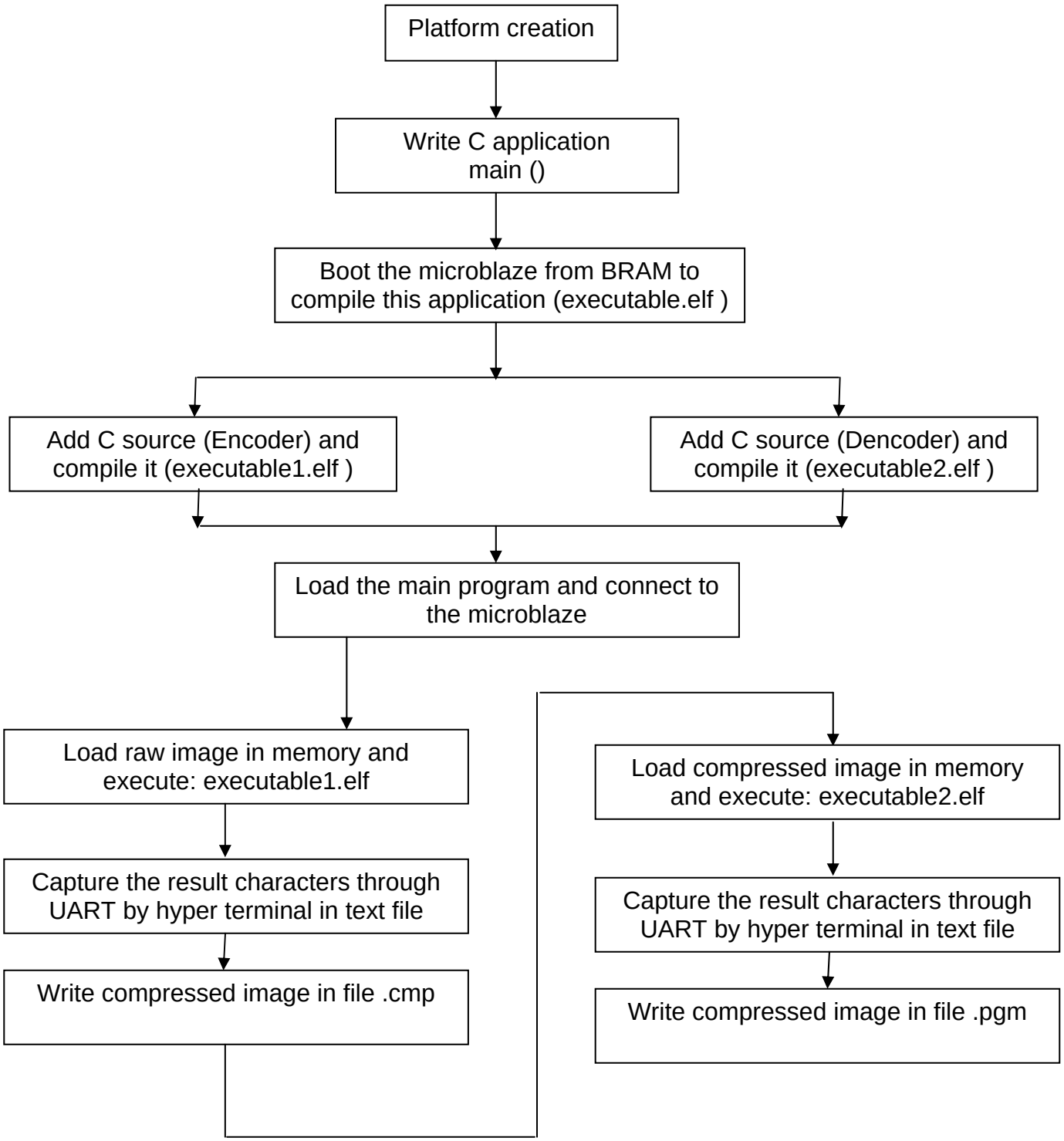


Figure 6.3: General Design Algorithm

The over all Steps :

1- ***Platform creation : Microblaze + UART + SDRAM + GPIO + MDM***
2- ***Write a small program to be executed by the microblaze on the BRAM : the microblaze boots from the BRAM. The created compiled program is : executable.elf.***
3- ***Write the C program for encoding (add source): compress.c. compiled under the name : executable1.elf.***
4- ***Execution :***

4 -1- Load the executable main program using the EDK procedures.

4 - 2- Run the XMD window, and go to the directory : mblaze and the subdirectory : code.

4 - 3- Connect to the microblaze using : connect mb mdm

Mdm: microprocessor debug module.

4 – 4 - Load the original image (data) using :

xdownload 0 –data image.pgm 0x8d000000

4 - 5- Run the encoding program using : Xdownload 0 executable1.elf

4 - 6- Run the hyper terminal.

4 -7- Run the command : text capture, in the hyper terminal.

4 - 8- Write the compressed image in the file: image.cmp, using the command :

Xcontinue 0 0x8c000000

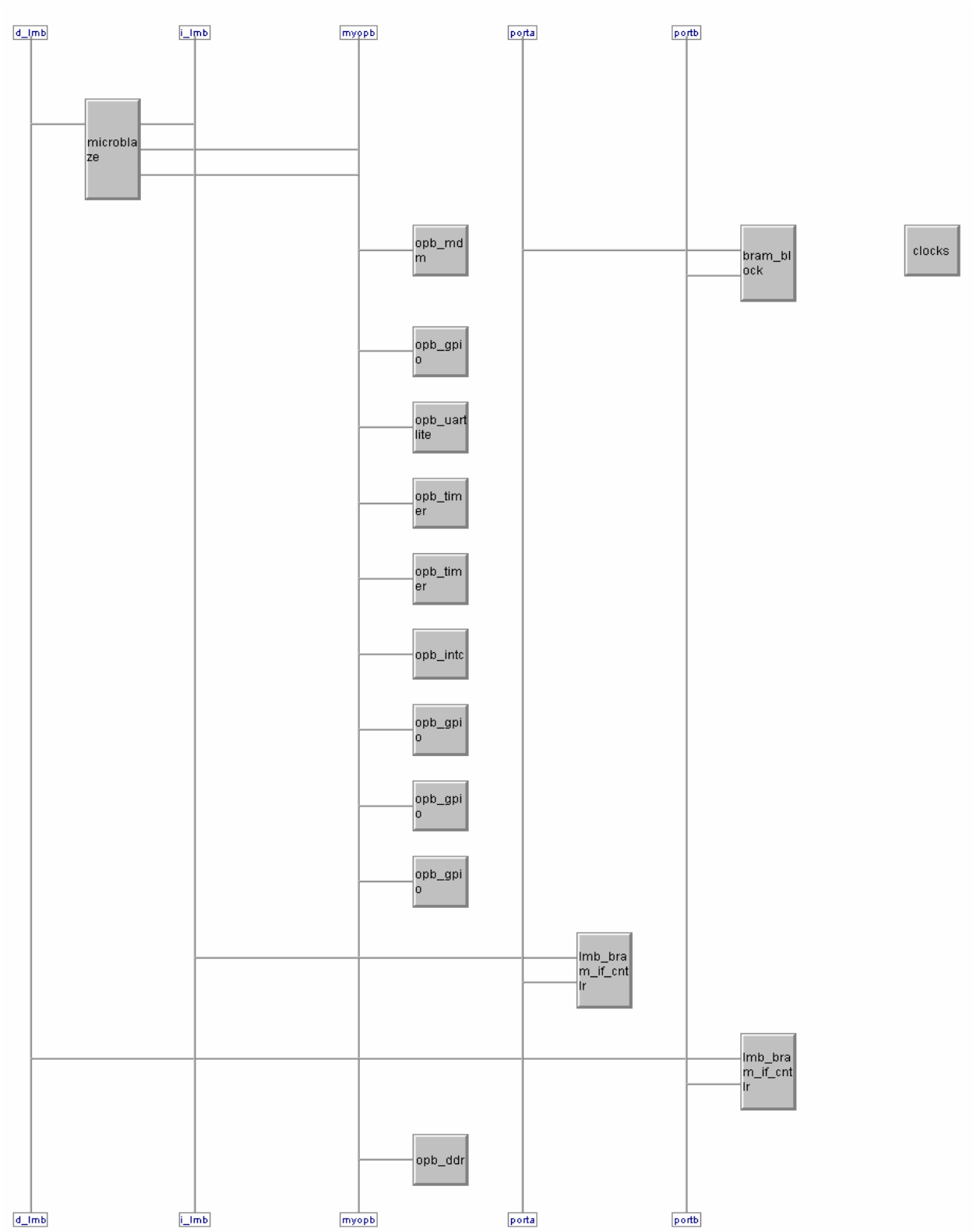


Figure 6.4: The created hardware plate form

Encoding steps :

1- Read the raw image : the image is stored in the SDRAM.

2 - Execute the wavelet transform function : forword wavelet.

3 - Quantize the wavelet coefficients.

4 - Encode the created zeros from quantization using the Run Length Encoder (RLE).

5 - Encode the significant coefficients using Huffman code.

6 - Write the compressed image: transmit resulted characters towards the UART, these characters are captured by the hyper terminal and written in a text file.

Decoding steps :

1- Read compressed image: the data (compressed image) is stored in the memory SDRAM.

2 – Decode the characters using Huffman inverse code.

3 – Add the counted zeros using the RLE decoder.

4 - Dequantize the wavelet quantized coefficients.

5 - Execute the inverse wavelet transform function: reverse wavelet.

6 - Write the decompressed image: transmit resulted characters towards the UART, these characters are captured by the hyper terminal and written in a file (decompressed

6.6 - Design Summary

In this chapter we have created a mixed hardware/software application in which the software portion communicates directly with the hardware process using streams-based communications. We have shown the steps needed to generate the FPGA hardware, combine this with a soft processor and download the combined application to an actual FPGA prototyping board.

This process has allowed us to determine the correctness of the algorithm in hardware and set up an in-system test environment to directly evaluate the results of subsequent design optimizations. In effect, what we have created by doing this is a hardware/software test bench that supports unit testing of this particular algorithm. This test bench lets us validate the algorithm quickly and apply a wide variety of test inputs simply by changing and recompiling the software application, which runs on the FPGA's embedded processor.

Armed with this software/hardware test bench, we can now proceed to optimize this application for performance, secure in the knowledge that the results will be fully testable at the process level, in actual hardware.

Timing is an important metric when comparing the hardware and software implementation. The timing results of image encoding algorithm executed on Xilinx Vertex-II FPGA (frequency:100MHz) and on Pentium IV (frequency:3300MHz); indicate that for both 256X256 and 512X512 type images, execution time is more better in the PC but the embedding and low cost characteristics are of the FPGA make it the better choice for our application. In figures (Fig 6.5 and Fig 6.6), different standard test images are used in the laboratory to determine compression metrics (Compression ratio, root mean square error and pic signal to noise ratio). These images are encoded by the FPGA, collected in a host computer database, transmitted through a local network and decoded in a distant computer. The root mean square error increases with the presence of neighbouring abrupt changes in grey level values (image 2 more than image 1, in figure 6.5), and with the presence of many grey levels in one image (image 3 more than image 4, in figure 6.6).

We can see also, for the pic signal to noise ratio (PSNR), that this factor increases with decreasing RMSE, see table 6.1 and figure 6.9.

Figure 6.5: Original images 1,2 (with shadow), level-3 wavelet transform and the decoded images from the encoded and transmitted images through a local network

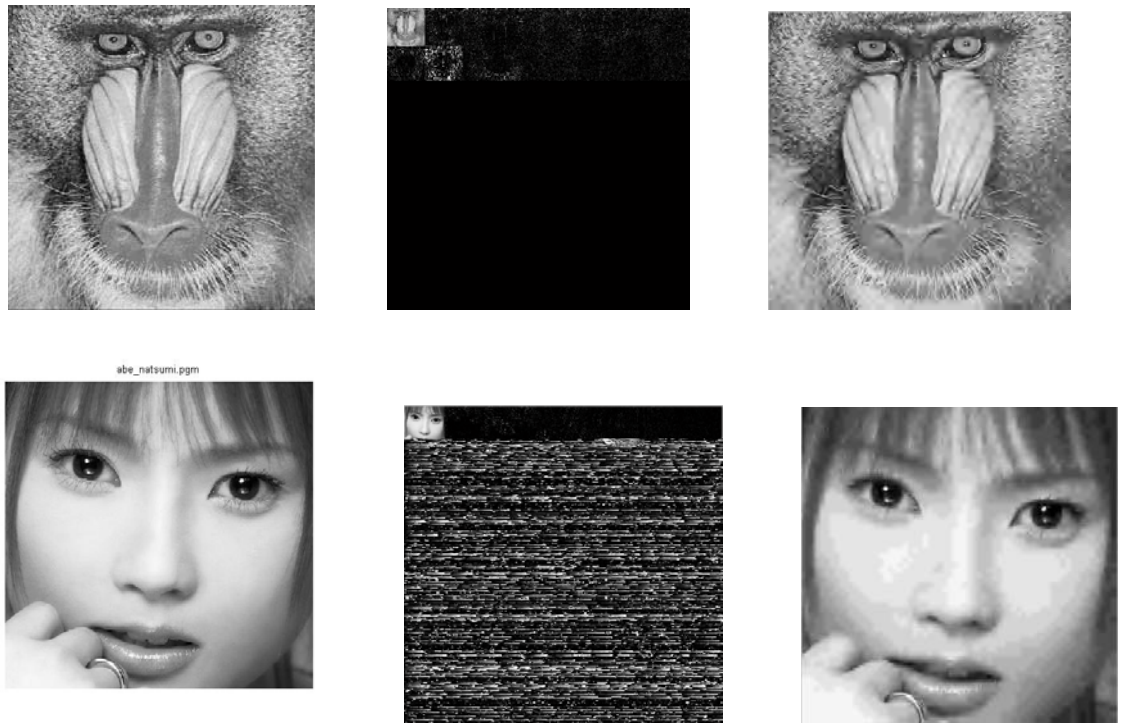


Figure 6.6: Original images 3,4 (with neighbouring changes for the first and less changes for the second), level-3 wavelet transform and the decoded images from the encoded and transmitted images through a local network

6.7 – Experiment and Results:

1- The experiment in the site was conducted by functioning the nuclear Reactor at a power of 1MW for 1 hour (This level of power and duration are used by physists in neutron radiography).

2- One operator near the horizontal channel is manipulating objects subjected to neutron radiography. (This work will be remote controlled in the tomography installation using a rotating table).

3- Images are collected in database personal computer (This will be accomplished by connecting the CCD camera to PC in the tomography installation), transferred sequentially to the FPGA board and retrieved through the serial communication cable and transmitted to a distant personal computer to be explored by physists after decompression.

4- By imaging many objects under test, for different applications, ranging from industrial to research purposes, we got the following encouraged results, figures 6.7 and 6.8, from which can assess our implementation on FPGA as profitable regarding both the proven advantages of software and the flexibility of hardware.

Finding a compromise between execution time and device usage is very important in hardware implementations, thus reaching up to 60% of device usage with a very short time for execution (less than milliseconds) is a very good job, suppose that the FPGA board is totally dedicated to this application.

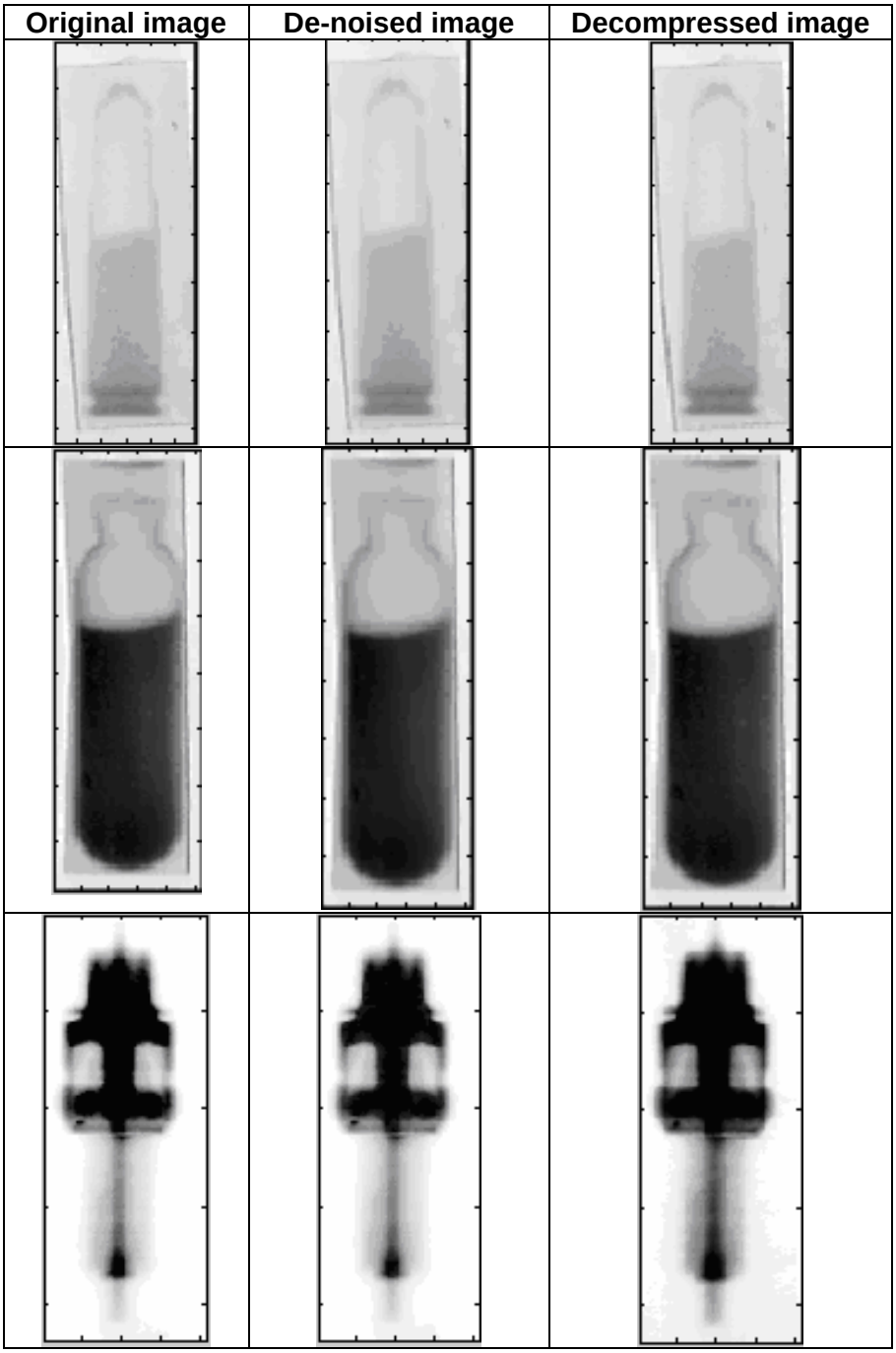


Figure 6.7: Some Neutron radiography experiments in pyrotechnics and electrical industry

Figure 6.8: Mixing light water (H_2O) with heavy water (D_2O)

IMAGE	***RMSE***	***PSNR***
Image 1	9.4566	28.6161
Image 2	19.0061	22.5529
Image 3	25.7113	19.9283
Image 4	7.6048	30.5090
Image 5	7.2025	30.9812
Image 6	6.6343	31.6949
Image 7	7.0790	31.1314
Image 8	7.8692	30.2122
Image 9	7.3402	30.8166
Image 10	8.1509	26.7304

Table 6.1: Evolution of the root mean square error and the pic signal to noise ratio (PSNR) with different images

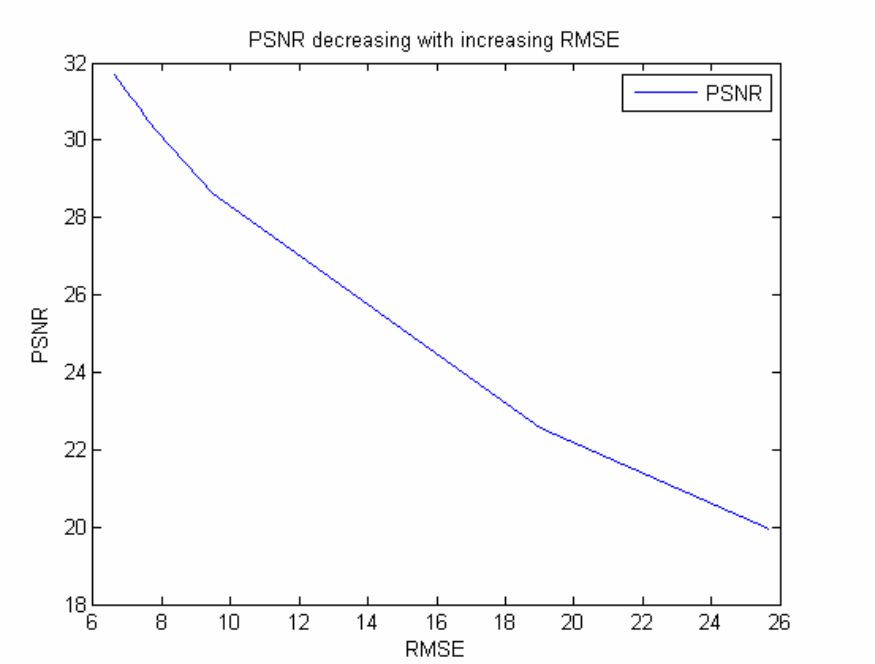


Figure 6.9: The PSNR in function with the RMS

CONCLUSIONS AND FUTURE WORK

The ongoing growing success in image processing, which has been characterised by the adoption of some wavelet-based schemes, is due to features inherent to the transform, such as time-scale localisation and multiresolution capabilities. In this memory, the basic concepts of the wavelet transform have been introduced and its features with the mathematical foundations behind it were reviewed. Depending on the application at hand, Mallat's algorithm for implementing the wavelet transform have been developed.

The emergence of high density, high performance FPGAs provides an attractive alternative for implementing complex signal and image processing applications. They provide signal and image processing designers with the flexibility of software while keeping intact the hardware performances. In this memory, benefits of this new computing concept over traditional computing have also been highlighted. Even though FPGA devices are supplied by different manufacturers, they share several features. The architecture of Xilinx Virtex II FPGA (used in this memory) have been described.

New methods and techniques are nowadays available which have the potential to improve the image quality and investigation capabilities in neutron tomography as well as radio isotopic imaging. For analyzing methods used around the future Es-Salam neutron tomography facility, our approach in image compression helps a lot for 2D image storage and transmission. It is based on projection images collection, encoding/decoding, TCP/IP transmission and therefore conversion to suitable format for 3D reconstruction.

A software and hardware implementations of the whole Encoder/Decoder system have been carried out, tested and presented in this memory. We gave a comparison between software implementation on a personal computer and embedded hardware implementation on FPGA. This demonstrates the feasibility of the use of FPGA's in image compression. Results show that better results are obtained when implementing in C language. However, even though the C implementation is better, when analyzing results we can observe that if we had a board with larger memory available, we could possibly obtain better results, making the FPGA an adequate alternative for algorithm acceleration. The particular details on the FPGA board will influence the results obtained in the algorithm implementation.

The results of this work demonstrate that the main bottleneck that limits improving execution time is the board's memory capacity and data transfer. ASIC implementations are another alternative to accelerate the job. However, since the cost of producing an ASIC is large in terms of price and time, this alternative is not appropriate for algorithm testing and development. FPGAs are simple to be reprogrammed, their configuration can be easily changed, and they are cost effective.

Concerning obtained encoding/decoding results, physists working on neutron radiography in Es salem research reactor appreciated well the work done and recommended to integrate it in the tomography installation.

Future works will be concentrated on improving compression of large images from tomography CCD camera (ex: 1024X1024 pixels and 2048X2048 pixels), as well as developing other image processing applications, in the frame of our nuclear imaging system, using FPGA new features.

APPENDIX

Reports:

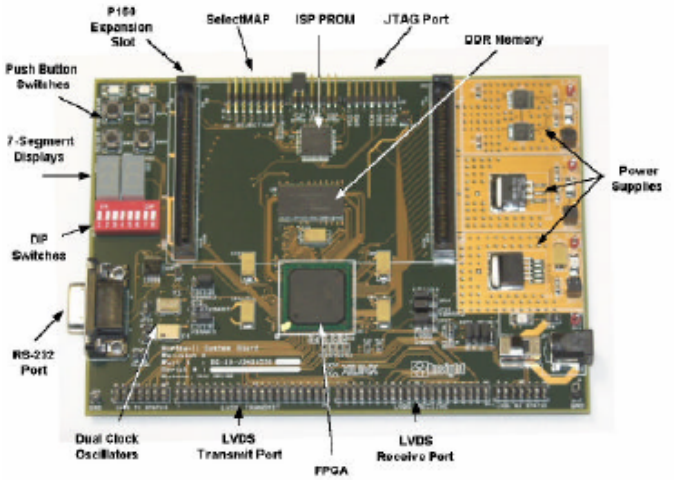
```
#------------------------------------------#
# Starting program ngdbuild
# ngdbuild -p xc2v1000fg456-4 -nt timestamp -bm system.bmm
C:/ucosjpeg/implementation/system.ngc -uc system.ucf system.ngd
#------------------------------------------#
Release 6.3i - ngdbuild G.35
Copyright (c) 1995-2004 Xilinx, Inc.  All rights reserved.
PM_SPEC -- Xilinx path component is <C:/EDK63>

Command Line: ngdbuild -p xc2v1000fg456-4 -nt timestamp -bm system.bmm -uc
system.ucf C:/ucosjpeg/implementation/system.ngc system.ngd
Reading NGO file "C:/ucosjpeg/implementation/system.ngc" ...
Reading component libraries for design expansion...

NGDBUILD Design Results Summary:
Number of errors:     0
Number of warnings: 338
Total memory usage is 61848 kilobytes
Writing NGD file "system.ngd" ...
Writing NGDBUILD log file "system.bld"...
NGDBUILD done.


#------------------------------------------#
# Starting program map
# map -o system_map.ncd system.ngd system.pcf
#------------------------------------------#
Release 6.3i - Map G.35
Copyright (c) 1995-2004 Xilinx, Inc.  All rights reserved.
PM_SPEC -- Xilinx path component is <C:/EDK63>

Using target part "2v1000fg456-4".
Removing unused or disabled logic...
Running cover...
Writing file system_map.ngm...
Running directed packing...
Running delay-based LUT packing...
Running related packing...
Writing design file "system_map.ncd"...

Design Summary:
Number of errors:      0
Number of warnings:  102
Logic Utilization:
  Number of Slice Flip Flops:       2,144 out of  10,240   20%
  Number of 4 input LUTs:           2,414 out of  10,240   23%
Logic Distribution:
  Number of occupied Slices:        2,068 out of   5,120   40%
  Number of Slices containing only related logic:   2,068 out of   2,068  100%
  Number of Slices containing unrelated logic:          0 out of   2,068    0%
        *See NOTES below for an explanation of the effects of unrelated logic
Total Number 4 input LUTs:          2,958 out of  10,240   28%
  Number used as logic:             2,414
```

```
  Number used as a route-thru:         63
  Number used for Dual Port RAMs:     320
    (Two LUTs used per Dual Port RAM)
  Number used as Shift registers:     161

  Number of bonded IOBs:               72 out of     324   22%
    IOB Flip Flops:                    71
    IOB Dual-Data Rate Flops:          23
  Number of Block RAMs:                32 out of      40   80%
  Number of MULT18X18s:                 3 out of      40    7%
  Number of GCLKs:                      5 out of      16   31%
  Number of DCMs:                       2 out of       8   25%
  Number of BSCANs:                     1 out of       1  100%

   Number of RPM macros:            5
Total equivalent gate count for design:  2,210,840
Additional JTAG gate count for IOBs:  3,456
Peak Memory Usage:  116 MB

NOTES:
   Related logic is defined as being logic that shares connectivity -
   e.g. two LUTs are "related" if they share common inputs.
   When assembling slices, Map gives priority to combine logic that
   is related.  Doing so results in the best timing performance.

   Unrelated logic shares no connectivity.  Map will only begin
   packing unrelated logic into a slice once 99% of the slices are
   occupied through related logic packing.

   Note that once logic distribution reaches the 99% level through
   related logic packing, this does not mean the device is completely
   utilized.  Unrelated logic packing will then begin, continuing until
   all usable LUTs and FFs are occupied.  Depending on your timing
   budget, increased levels of unrelated logic packing may adversely
   affect the overall timing performance of your design.

Mapping completed.
See MAP report file "system_map.mrp" for details.


#------------------------------------------#
# Starting program par
# par -w -ol std system_map.ncd system.ncd system.pcf
#------------------------------------------#
Release 6.3i - Par G.35
Copyright (c) 1995-2004 Xilinx, Inc.  All rights reserved.
PM_SPEC -- Xilinx path component is <C:/EDK63>
Constraints file: system.pcf
Loading device database for application Par from file "system_map.ncd".
   "system" is an NCD, version 2.38, device xc2v1000, package fg456, speed -4
Loading device for application Par from file '2v1000.nph' in environment
C:/Xilinx.
The STEPPING level for this design is 0.
Device speed data version:  PRODUCTION 1.118 2004-06-25.


Resolving physical constraints.
```

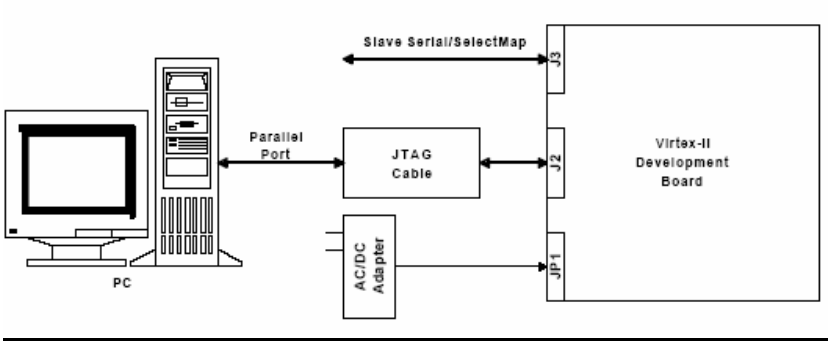
```
Finished resolving physical constraints.

Device utilization summary:

   Number of External IOBs            72 out of 324    22%
      Number of LOCed External IOBs   69 out of 72     95%

   Number of MULT18X18s                3 out of 40      7%
   Number of RAMB16s                  32 out of 40     80%
   Number of SLICEs                 2068 out of 5120   40%

   Number of BSCANs                    1 out of 1     100%
   Number of BUFGMUXs                  5 out of 16     31%
   Number of DCMs                      2 out of 8      25%

Writing design to file system.ncd.
Total REAL time to Placer completion: 41 secs
Total CPU time to Placer completion: 38 secs
Total REAL time to Router completion: 4 mins 34 secs
Total CPU time to Router completion: 4 mins 29 secs
Total REAL time to PAR completion: 4 mins 40 secs
Total CPU time to PAR completion: 4 mins 35 secs

All signals are completely routed.

**************************
Generating Clock Report
**************************
```

Clock Net	Resource	Locked	Fanout	Net Skew(ns)	Max Delay(ns)
porta_BRAM_Clk	BUFGMUX7P	No	1476	0.465	1.379
DBG_CLK_s	BUFGMUX5S	No	140	0.108	1.183
ddr_clk_90_s	BUFGMUX1P	No	84	0.173	1.180
sys_clk_90_s	BUFGMUX0S	No	10	0.088	1.019
sys_clk_raw_IBUF	Local		3	0.000	1.784
opb_mdm_0/bscan_update	Local		1	0.000	0.782

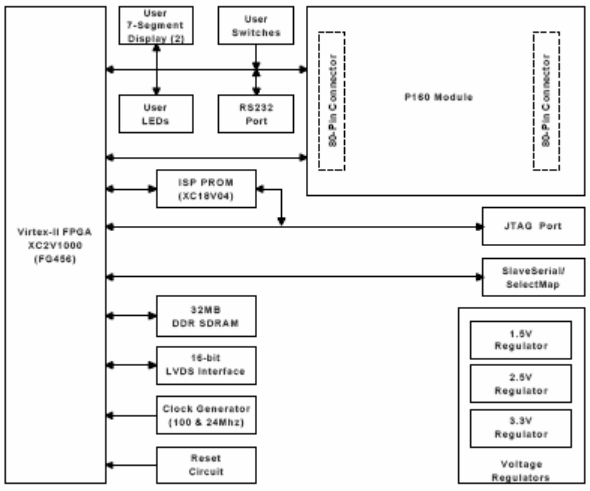
```
Writing design to file system.ncd.
PAR done.

#------------------------------------------------#
# Starting program post_par_trce
# trce -e 3 -xml system.twx system.ncd system.pcf
#------------------------------------------------#
Release 6.3i - Trace G.35
Copyright (c) 1995-2004 Xilinx, Inc.  All rights reserved.

Design statistics:
   Minimum period:  12.751ns (Maximum frequency:  78.425MHz)
   Maximum path delay from/to any node:   2.295ns
Analysis completed Sat Apr 14 15:28:11 2007
Generating Report ...
Total time: 8 secs
```



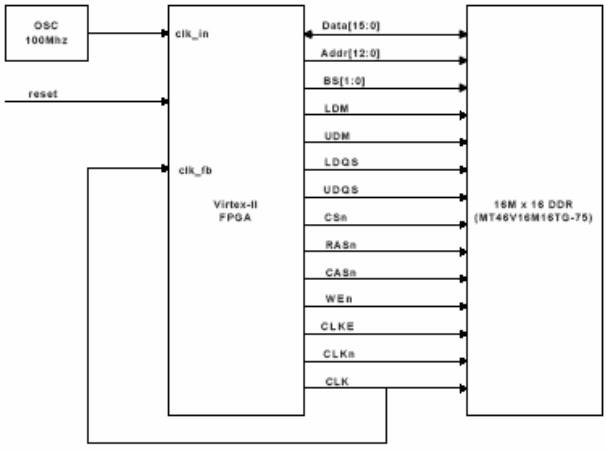
Virtex-II system Board



Download Setup



Virtex-II System Board Block Diagram



DDR Interface

Scientific Participations in relation with the memory :

1: S.Saadi, M.Touiza, A.Guessoum, “FPGA implementation of 2D signals Encoder using QMF based dyadic DWT: application to neutron tomography projections”, ***published in***: Journal of Applied Sciences, ANSI journals, N: 254-JAS-DOI 2007. ISSN 1812-5654.

2: S.Saadi, M.Touiza, A.Guessoum, “Embedded implementation of 2D signals Encoder/Decoder using QMF based dyadic DWT fast wavelet MALLAT’s Algorithm: Application to nuclear imaging”, ***accepted*** for oral presentation at International Symposium on Signals, Systems, and Electronics (ISSSE 2007), July 30 - August 2, 2007, Montréal, Québec, Canada.

3: S.Saadi, M.Touiza, A.Guessoum, “FPGA implementation of 2D signals Encoder/Decoder using QMF based dyadic DWT fast wavelet MALLAT’s Algorithm: Application to Radioisotopes images and neutron Tomography Projections”, ***accepted*** for oral presentation at International Conference on Electrical Engineering Design and Technologies (ICEEDT), November 5-6, 2007 Hammamet, Tunisia.

4: S.Saadi, M.Touiza, A.Guessoum, “ Embedded system for image encoder/decoder based on fast wavelet mallat’s algorithm: application to nuclear imaging “, ***accepted*** for oral presentation at the International Conference on Computer Integrated Manufacturing CIP’2007, November, 03 – 04, 2007 Setif, Algeria. Ferhat ABBAS University, Laboratory of Automatic and IEEE.

5: S.Saadi, M.Touiza, A.Guessoum, “FPGA implementation of 1D/2D signals Encoder/Decoder using QMF based dyadic DWT: application to neutron tomography projections”, ***accepted*** for oral presentation at the International Conference on Modeling and Simulation, MS’07, Algiers, July 2-4,2007.

References

1: S.K.Knapp, "*Using Programmable Logic to Accelerate DSP functions*" Xilinx application note, 1995. (Available at www.xilinx.com).

2: Uwe Meyer-Baese, "*Digital Signal Processing with Field Programmable Gate Array*",Springer Verlag Berlin Heidelberg New York,2001, http://www.Springer.de

3: W.R.Rhines, "*Discontinuity at the Gate: A New Era in FPGA Design*", Xcell Journal, Issue 40, pp.8-10, 2001.

4: R.G.Goslin, "*Using Xilinx FPGAs to Design Custom Digital Signal Processing Devices*", Xilinx application notes, 1999.

5: Stephen Brown and Zvonko Vranesic,"*Fundamentals of Digital Logic with VHDL Design*", second edition,2005,Mc Graw Hill,Department of Electrical and Computer Engineering,University of Toronto, www.mhhe.com.

6: Xilinx, "*Virtex II Pro Platform FPGA Handbook*", Ver 1.0, January 2002 (available at www.xilinx.com).

7: M.Siani,"*Platform FPGAs take on ASIC SOCs*", Xcell Journal, Issue 42, pp. 35-38, March-April 2002.

8: C.S.Burrus, r.a.Gopinath and H.Guo, «*Introduction to Wavelets and Wavelet Transforms: A primer* », Prentice Hall, 1998.

9: S.Mallat, "*A theory for multiresolution signal decomposition: the wavelet representation*", IEEE Transactions on Pattern Recognition and Machine Intelligence, Vol.11, No.7, pp. 674-693, July 1989.

10: A.L. Graps, "*An introduction to Wavelets*", IEEE Computational Science and Engineering, pp 50-61, 1995.

11: www.xilinx.com/ise/embedded/edk_docs.htm.

12: T. Ebrahimi, C. Christopoulos and D.T. Lee (Eds), " *JPEG2000* ", Special Issue of Signal Processing: Image Communication Vol. 17, Issue 1, Elsevier Science, Jan 2002.

13: J.L. Starck, F. Murtagh and A.Bijaoui, "*Image Processing and Data Analysis*", Cambridge University Press, 1998.

14: A.Grossman and J.Morlet, "*Decomposition of hardy functions into square integrable wavelets of constant shape*", SIAM Journal of Mathematical Analysis, Vol.15, pp.723-736, 1984.

15: W.Sweldens, *" Wavelets: what next? "*, Proceedings of the IEEE, 84(4), pp 680-685, 1996.

16: I.Daubechies, " *Ten lectures on Wavelets*", SIAM, Philadelphia, 1992.

17: N.J.Fliege, "*Multirate Digital Signal Processing*", John Wiley, 1994.

18: G.Strang and T.Nguyen, "*Wavelets and Filter Banks*", Wellesley-Cambridge Press, 1996.

19: S.Masud, "*VLSI system for Discrete Wavelet Transforms*", PhD thesis, Department of Electrical Engineering, The Queen's University of Belfast, Sep 1999.

20: John G.Proakis,Dimitris G.Manolakis*,"Digital Signal Processing: Principles, Algorithms, and Applications*", Third Edition,1996,Prentice Hall International, INC.

21: Stephen wong, Loren zaremba, David Gooden, and H. K. Huang*," Radiologic Image Compression-A Review",* Proceedings the IEEE, Vol. 83, No. 2, February 1995.

22: Benzid Redha*," Ondelettes et Statistiques d'Ordre Supérieur Appliquées aux Signaux Uni et Bidimensionnels",* Thèse Doctorat Es Science enElectronique.2005, Departemnt d'electronique, Faculté de l'ingénieur,Université de Batna, Algérie.

23: http://www.netrino.com/Articles/RCPrimer/)

24: www.altera.com

25: www.xilinx.com

26: K.Alotaibi, "*A High Level Hardware Description Environment for FPGA-based Image Processing Applications*", PhD thesis, School of Computer Science, The Queen's University of Belfast, May 1999.

27: IEEE, "*IEEE Standard 1076-1993, Standard VHDL Language Reference Manual*", 1994.

28: P. Moorby, "*History of Verilog*", IEEE Design & Test of Computers, Vol.9, No.3, pp.62-63, 1992.

29: U. Heinkel, W. Glauert, M. Padeffke and M. Wahl , "*The VHDL Reference*", John Wiley, April 2000.

30: Parimal Patel ,"*Embedded Systems Design Using FPGA*", Xilinx, Inc, San Jose, CA, USA. Proceedings of the 19th International Conference on VLSI Design (VLSID'06).

31: A. C. Kak and M. Slaney, "*Principles of Computerized Tomographic Imaging*", IEEE Press, New York (1998).

32: M. Schneider, "*Studies for Neutron Tomography at the Institute Laue-Langevin*", Thesis Diploma submitted at the Institute of Physics Heidelberg, University of Heidenberg (2001).

33: Maria Ines S. Souza, "*Comparaison of tomographic systems for X-ray and thermal neutrons*", Brazilian Journal of Physics, vol.33, n° 2 (2003).

34: Wade.J. Richard, "*Neutron Tomography developments and applications*", UCD McClellan Nuclear Radiation Center , University of California Davis (2003).

35: P.A. Lynn and W. Fuerst, "*Introductory Digital Signal Processing with Computer Applications*", John Wiley & Sons, 2000.

36: R. I. Hartley and K. K. Parhi, 'Digit-Serial Computation', Kluwer Academic, USA, 1995.

37: H.Suzuki, Y.N.Chang, K.K.Parhi, "*Performance Tradeoffs in Digit-Serial DSP Systems*", Proceeding of Asilomar Conference on Signals, Systems and Computers, November 1998, Pacific Grove, USA.

38: M.Nibouche, A.Bouridane, O.Nibouche, D.Crookes, "*Rapid Prototyping Of Orthonormal Discrete Wavelet Transforms On FPGAs*" Proceedings of the IEEE International Symposium on Circuit and Systems (ISCAS), Sydney, Australia, May 2001.

39: Z.Navabi, "*VHDL: Analysis and Modelling of Digital Systems*", McGraw Hill, 1993.

40: Rafael Gonzalez,"*Digital Image Processing Using Matlab*",3rd edition,2004,Printice Hall,Upper Saddle River.

41: David Pellerin, Scott Thibault, "*Practical FPGA Programming in C*", Prentice Hall PTR, ISBN: 0-13-154318-0, April 22, 2005.

42: PD Habib Zaidi, Ph.D , Division of Nuclear Medicine Geneva University Hospital, Switzerland "*Positron Emission Tomography: State of the art and future perspectives*", Algiers on 2006, May,21.