IMPLEMENTATION OF FILTERED BACKPROJECTION ALGORITHM ON FPGA

A DISSERTATION

Submitted in partial fulfillment of the
requirements for the award of the degree
of
MASTER OF TECHNOLOGY
in
ELECTRICAL ENGINEERING
(with Specialization in System Engineering and Operations Research)

By

SHASHANK SHARAN RAI

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE -247 667 (INDIA)
JUNE, 2007

I.D. No.-M.T. 412/2007-451/7 G

INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE

CANDIDATE'S DECLARATION

I hereby declare that the work, which is being presented in this dissertation entitled **IMPLEMENTATION OF FILTERED BACKPROJECTION ALGORITHM ON FPGA** in the partial fulfilment of the requirements for the award of the degree of **Master of Technology in Electrical Engineering** with specialization in **System Engineering and Operations Research**, submitted in the **Department of Electrical Engineering**, Indian Institute of Technology Roorkee, Roorkee is an authentic record of my own work carried out during a period from May 2006 to June 2007 under the supervision of **Dr. Indra Gupta**, Associate Professor, Electrical Engineering Department, Indian Institute of Technology Roorkee, Roorkee.

The matter presented in this thesis has not been submitted by me for the award of any other degree of this or any other Institute.

S.S.Rai

(SHASHANK SHARAN RAI)

This is to certify that the above statement made by the candidate is correct to the best of my knowledge and belief.

Indra

Date: 28/6/07

Dr. Indra Gupta
Associate Professor
Department of Electrical engineering
Indian Institute of Technology
Roorkee.

ABSTRACT

Recently, Field Programmable Gate Array (FPGA) technology has become a promising and viable platform for the implementation of image and video processing algorithms. The special features available with today's FPGAs have allowed the digital technology to be used in many such applications. Image reconstruction is a problem that involves complex trigonometric calculations and is inherently serial in nature. A high-speed hardware implementation of Filtered backprojection algorithm can address the problem of real-time image reconstruction for various practical applications.

Today most of the applications where image reconstruction is required are demanding for a real-time image processing facility. However, an inherent serial nature of the process makes it infeasible with a software implementation. Though the software implementation of an image-processing algorithm may be comparatively easy from the development point of view, an upper limit on the speed of the execution is imposed by the slow speed is imposed by the slow speed interfaces. In that case, even a very fast CPU is not able to deliver the desired performance. For such cases, a dedicated hardware implementation generally gives a better result.

As images grow larger in size and depth, the software implementation of image processing algorithms becomes less useful. For a real-time system, a hardware implementation may provide a more useful solution. In the current work, a Xilinx Virtex II-Pro video board is being used to implement the algorithm. A novel approach is proposed for the implementation of Filtered backprojection algorithm, which combines filtering and backprojection processes on a single chip. Few advanced features of Xilinx Virtex device are used to design architecture for Filtered backprojection algorithm that is able to perform filtering and backprojection operations in parallel. A one dimensional FIR filter has been implemented for filtering the individual projections. Then a custom digital circuitry is implementing the backprojection part of the algorithm. The final goal of the work is to demonstrate the reconstructed images from their respective Radon transforms.

ACKNOWLEDGEMENTS

It is my proud privilege to express my deep sense of gratitude and indebtedness towards my thesis supervisor Dr. Indra Gupta, Associate Professor, Department of Electrical Engineering, for her invaluable guidance & criticism, and kind and continuous encouragement, which were the vital factors in successful completion of the present work. I am heartily thanking her for her deep concern towards my academics and personal welfare.

Her painstaking support and involvement in preparation of manuscript, theoretical analysis, simulation, and hardware result studies are gratefully acknowledged. I humbly acknowledge a lifetime's gratitude to her and hope for a continued interaction even in the future.

I express my deep sense of gratitude to the Prof. S. P. Gupta, Head, Electrical Engineering Department, Indian Institute of Technology, Roorkee, for providing excellent facilities and nice working atmosphere in the department for the research work.

I am thankful to Prof. M. K. Vasantha in so many ways. His constant encouragement and willingness to listen to and help with my academic queries or personal problems are some of the things I benefited with. I found him a complete teacher, who taught me that it is equally important to be a good human being as much as to be a good technocrat. The education given by him will help me all my life.

My sincere thanks are also due to Dr. H. O. Gupta, Dr. Surendra Kumar, Dr. Rajendra Prasad, and Dr. G. N. Pillai for extending moral support and technical discussions as and when required during the work.

I thankfully appreciate and acknowledge my indebtedness to research scholars Dr. Vishal Kumar, Dr. Vijayender Singh, Mr. Rohit Bhakar, Mrs. Nidhi Singh, Mrs. Nidhi Kulkarni and Mr. S. K. Tomar for their help, cooperation, and moral support from time to time. My special thanks goes to the fellow classmates Mr. Rohith Kumar H. C., Mr. Soumyojoti Maitra, Ms. Sowmya Kollipara, Mr. Patel Vinod Kumar, Mr. Siluveru Karunakar, Mr. Srikanth Reddy, Mr. Anil Kumar, Mr. Kalyan Ayyagiri, Mr. M. Chandrashekar, Mr. Praveen Rangisetti for their cooperation and moral support during my stay.

Besides the enjoyment of doing research work, this time has also given me an opportunity to make great friends. I shall never be able to forget the time, which I spent with Mr. V.S.Sriram and Mr. Sheri Sundeep and this writing space is not enough to write about the memories I have in my mind.

I am thankful to the technical staff of the SEOR Lab Shri Kalyan Singh, Shri C.M. Joshi, and other staff members for their timely cooperation and needful help.

Thanks are also due to all those who helped me directly and indirectly for the completion of the work.

My heartiest gratitude goes to my parents, brothers and other members of family, for constantly supporting, which allowed me to concentrate on my work.

I would like to dedicate this research work to my family.

(Shashank Sharan Rai)

CONTENTS

Page No.

LIST OF FIGURES

LIST OF SYMBOLS

$p(x)$	Projection function
μ_i	Attenuation coefficient of ith tissue along the X-ray beam
I_0	Intensity of projected radiation
I	Intensity of received radiation
s	One coordinate in radon space, represents the distance of the line of projection from origin
θ	Second coordinate in radon space, represents the angle of projection
x	Horizontal coordinate in image space
y	Vertical coordinate in image space
$f(x,y)$	Two-dimensional object function, f is the intensity of function at (x, y)
L	Set of parallel lines of projection
r	Distance along a line of projection
$p_\theta(s)$	Projection, one-dimensional array of projection functions along the set of lines L at an angle θ
$p(s, \theta)$	The Radon transform
$\overline{p}\ (s, \theta)$	Discrete Radon Transform
ω	Natural frequency component in frequency domain
$S_\theta(\omega)$	One-dimensional Fourier transform of $p_\theta(s)$
$F(u,v)$	Two-dimensional Fourier transform of $f(x, y)$
(u,v)	Fourier domain coordinates
$F(\omega, \theta)$	A slice in Fourier domain corrpesponding to $S_\theta(\omega)$
d	Dimension of the image
N	Number of projections
P	Pixel array
g	Multiplication coefficients in Iterative reconstruction
bin	Rounded integer address in the Radon space
fract	Interpolation factor used to compensate the rounding during projection and backprojection

LIST OF ABBREVATIONS

ART	Algebraic Reconstruction Technique
AIT	Active Interconnect Technology
ASIC	Application Specific Integrated Circuits
CAD	Computer Added Design
CMOS	Complementary Metal Oxide Semiconductor
CLB	Configurable Logic Block
CT	Computed Tomography
CAT	Computed Axial tomography
DCM	Digital Clock Manager
DRT	Discrete Radon Transform
DSP	Digital Signal Processing/Processor
DTFT	Discrete Time Fourier Transform
FBP	Filtered backprojection
FIR	Finite Impulse Response
FFT	Fast Fourier Transform
FPGA	Field Programmable Gate Array
FSM	Finite State Machine
GRM	General Routing Matrix
HDL	Hardware Description Language
IP	Intellectual Property
NDE	Non Destructive Evaluation
NRE	Non Recurring Engineering
PCB	Printed Circuit Board
RAM	Random Access Memory
RISC	Reduced Instruction Set Computer
ROM	Read Only Memory
RT	Radon Transform
SB	Switch Bank

SRAM	Static Random Access Memory
VLSI	Very Large Scale Integration
VHDL	VHSIC (Very High Speed Integrated circuit) Hardware Description Language

CHAPTER - 1

INTRODUCTION

1.1 GENERAL

Image reconstruction is a vital part of image processing, which appears in the cases where visual details of a section of opaque object are intended to be imaged. Normally, this object is a living body that can not be harmed for gathering the required information. Therefore, the required information is collected near the body of object rather than from inside. This data is collected by illuminating the object by penetrating radiations and measuring them after penetration, and is called projection or sinogram. From the projections taken at various angles, image of a desired section of the object can be obtained using an image reconstruction technique. The way, in which the projection data has been collected, decides the image reconstruction technique to be used. This type of imaging is called tomography [1]-[3]. Literal meaning of Tomography is imaging by sections or "slices".

The problem of reconstruction of the image, from measurements of its projections belongs to a class of inverse problems. An inverse problem is characterized by the fact that the information required to user is not directly available. In case of medical imaging, these projections are obtained by measurement of some type of electromagnetic radiation near the body of the patient, and the imaging device (camera or detector array) provides a transformation of the desired data. Therefore, for reconstructing the original image from available data, it is necessary to have knowledge of the mathematical operations performed by the imaging system.

J. Radon gave the mathematical principles of computed tomography (CT) [1]-[3] in 1917. At that time, his theory remained of mathematical interest only due to the lack of further research in that field and support of computation devices. His work was again used and extended over the complex fields by Kirillov during late 1950s. Although today, computed tomography is one of the most famous tools in medical diagnosis; its first application was found in field of radio astronomy during late 1950s. Presently, computed tomography is being used over diverse fields such as: medical imaging, seismology, radio astronomy, electron microscopy, and under water acoustic imaging. CT is used for non-invasive diagnostics, surgical planning etc., as a diagnostic tool. The idea behind tomography is to reconstruct an object from the integration of data on a slice-by-slice basis. Hence, a 3-dimensional object can be viewed in two different ways, as a stack of 2-dimensional slices or a natural 3-dimensional object. Original theory of CT defines the mathematical relationships between the object attributes and projections obtained using parallel beam geometry. Now, CT systems have gone through a series of generations, and the mathematical relations have been modified. A brief discussion about these generation has been given in following chapter of this thesis.

Prior to the introduction of digital computers, the problem of image reconstruction was limited to mathematical practices only. Before introduction of computers as an integral part of imaging systems, sectional images were created by longitudinal or transverse scanning. Conventional tomographic devices were able to image the whole anatomy as a slice. However, in this imaging system, the resulting image was poor in visual qualities because of the blurring effects caused by neighboring structures. Another contributing factor for poor image quality was shadowing caused by other structures of different density distributions falling in the same slice of the anatomy. However, even at this time, when the mathematical problem of image reconstruction has been explored very deeply, and being applied to many more fields other than medical imaging, there is still much debate about the most suitable numerical technique for image reconstruction.

Earlier CT scanners used iterative reconstruction techniques to reconstruct the images. Later on Fourier transform based techniques gained their importance after the introduction of "center-slice theorem" [1]-[4] and nowadays it forms the basis for most of the image reconstruction techniques. After Fourier based methods, backprojection based techniques appeared and received a wide acceptance because of their better accuracy and faster implementation. Hence, today the backprojection-based algorithms are the most favorable choice for real-time image processing.

The natural transform caused by a CT scanner is known as Radon transform (RT) [1]-[3]. The straightforward way to get the image from this data is to take inverse Radon transform. However, direct inversion of RT is not possible, as it have no non-trivial analog in the one-dimensional space. Then, a discrete version of RT (DRT) was suggested by Beylkin to simplify the inversion problem [12]. After the formulation of DRT, many ideas to implement RT using digital signal processor (DSP) chips and Very Large Scale Integration (VLSI) came into picture [10], [11]. Few more efforts to solve this projection based image-processing problem were also made during 1990s [6]-[9]. Later Matus and Flusser suggested two more discrete approximations to the classical RT [13].

Kingston and Svalbe [14], proposed a method to map the digital and continuous projections in Fourier space. Recently many works has been cited to be discussing about the fast implementation of Radon transform and its inverse on digital hardware [5], [15], [16]. Most of these works had assumed that the input data has been collected in advance and then it can be provided in parallel at the inputs and hence give a fast implementation [16]. For implementation of filtered backprojection, in most of the cited works filtering and backprojection are performed separately and performance analysis for only backprojection operation is compared. In the current work, an effort has been made to use the most recent features available with FPGAs to perform filtering and backprojection operations together. However, the performance of current design can not be directly compared with any cited work because the hardware constrains are restricting the implementation to be serial in nature. Although emphasis has been placed to achieve an improved performance in terms of speed and resource utilization.

In most of the computer assisted imaging systems, the reconstruction work is carried out by software implementation of a suitable image reconstruction algorithm. Today medical diagnostic and surgery demand for a real-time visualization of the object, which is not possible with a software implementation. An obvious reason is that the amount of data to be processed is large and the classical algorithms are inherently complex and serial in nature [5]. Moreover, this data needs to be processed repeatedly, which makes the software implementation a further bad choice. The projection data is stored in physical memory in a serial manner, but the addressing of the data from the algorithm is complex [25]. Hence, most of the times it may happen that the required data is not available in cache memory of the computer, and it has to be read from secondary storage memory like hard disk drive. Therefore, the slow speed memory devices further limit the speed of processing. These reasons make the image reconstruction using a software implementation a very time consuming and expensive proposition.

On the other hand, the real-time image processing needs filtering and reconstruction of the image within a short duration. The capability of real-time image processing helps in reducing the motion artifact in image and capturing the transient movements inside the object. Real-time imaging is also very much helpful in radiation therapy and other new techniques of surgery. Here it enables a doctor to monitor the region of interest during therapy or surgical procedure. Therefore, it is necessary to design a powerful image processing system, which can perform the required task at the desired speed. Until date, many efforts have been made for designing such a powerful computation system. The first idea was to design a multiprocessor system using several general-purpose processors. However, even the most powerful processors are designed to be general purpose, and hence somewhere or other their capacity to perform such complex data processing operations is sacrificed against their flexibility. Even if such a system is able to meet the processing requirements, the slow communication overheads limit their speed, and this condition would worsen as the system scale increases.

Another choice is to design a dedicated hardware for the particular operation. There are two options available for a designer, one is to design an application specific integrated circuit (ASIC) and second is to use a reprogrammable hardware. ASICs are the most efficient way to meet such speed requirements but they lack in flexibility. In the face of any change in algorithm or reuse of existing design, they cannot be adopted. Another factor is the design and development cost of an ASIC is very high, and it is still growing rapidly even with the shrinking size of semiconductor devices. However, reprogrammable devices offer a good relative trade-off between performance and flexibility. They are also affordable and promising than ASICs for a real-time image processing application design.

FPGAs are a class of re-programmable devices. FPGA contains a number of configurable logic cells or logic elements, which can be programmed to perform any basic operation of a digital design. Moreover, it also contains programmable interconnects which can either automatically used to connect the different logic cells or can be defined by designer to work as a data bus. Both of these features can be used together to design more complex digital units like state machines, signal processors or even a microprocessor. Many advanced FPGAs also provide additional facilities like on-chip Static Random Access Memory (SRAM), dedicated circuits like multipliers etc to simplify the design of a higher end digital device. SRAM-based FPGAs are said to be of a very long life as they can be re-programmed as many times as designer want [24].

Some advanced FPGAs also provide on-chip microprocessors like PowerPC-405 by Xilinx. Those FPGAs are called platform FPGAs and they provide another possibility to use the advantages of software-hardware co-design. While working with a platform FPGA, designer has an option to split the design in software and hardware and implement it to get the best possible results. Hence, nowadays FPGA are self-contained for a large SOC (system-on-chip) design [24]. Moreover, in medical imaging the power of FPGA lies in its flexibility, customizability and re-configurability. Here, customizability means FPGA can be programmed to perform a particular operation, as the algorithm requires it. Re-configurability means that an FPGA design can be altered or completely changed in case of any modification in the algorithm or if a new algorithm comes out.

1.2 OBJECTIVE OF THE DISSERTATION

The key objective of the present work includes studying of the image reconstruction problem from the medical imaging point of view, exploring the mathematical principles involved in computing projections and backprojection and development of a customized digital circuit to perform these operations on chip. Following points are given to point out the attempts made in the current work:

1. A comprehensive literature survey was conducted on computed tomography. Initially, different ways to perform the computed tomography were studied for understanding the system and its uses. Then, special attention was given to parallel beam tomography because this method forms the basis for all other geometries of tomography. Related principles and underlying mathematics in parallel beam tomography were looked deeply. Extensive studies were carried out to look for the best possibilities, which can be depicted for design and verification of parallel beam backprojection.
2. Since most of the standard mathematical tools available use, their own representation of data the first necessity was to generate the Radon transform [1]-[3] for the test images. Hence, another literature survey was made for the understanding Radon transform and its discretization issue in-depth. The Radon transform was implemented on FPGA for a better density. An additional issue of address generation in radon space was also addressed on-chip. For this purpose an Intellectual Property (IP) core provided by Xilinx Core-Generator was used.
3. Backprojection circuit has been designed with the same addressing strategy and every projection is backprojected on a grid of initially blank image. The results of backprojection hardware are in confirmation with the input image. An effort was made to keep the resource utilization low by using a fixed-point arithmetic.
4. Filtered Backprojection (FBP) algorithm has been implemented on FPGA chip. In this design on-chip memory resource (Block SRAM) has been used to store intermediate data. The design is also having a special feature that allows it to

perform filtering and Backprojection operation simultaneously with a small time delay.

VHDL (Very high speed integrated circuit Hardware Description Language) is used here to implement the design. Among all the available HDL language, VHDL was used because of it is easy to design large digital structures using smaller modules in VHDL. Also VHDL is most widely used HDL; hence, the code can be understood and reused easily if required later.

1.3 ORGANIZATION OF THESIS

The organization and flow of rest of work is explained under many chapters as following:

Chapter 2 presents a detailed overview of the computed tomography process, defines the various terminologies, which must be known before going through next chapters. In addition, it presents some basic mathematics involved in CT.

Chapter 3 includes the discussion of basic principles for image reconstruction from projections. Several techniques utilized for image reconstruction from projections are discussed in this chapter. Finally, FBP (Filtered Backprojection) is discussed both in continuous and discrete space. Need of filtering and type of filtering required are discussed here.

Chapter 4 this chapter presents the formulation of algorithms for all problems, which has been implemented on the FPGA. The assumptions made, truncation and discretization used in formulation of those algorithms are discussed.

Chapter 5 explains the design flow on an FPGA device and explains various terminologies involved in it.

Chapter 6 this chapter shows the practical results obtained after placing the design on FPGA chip. Simulation results for some modules are included to give a better understanding of underlying complex digital functions.

Chapter 7 presents the discussion regarding the improvements and advancements that can be made in the current work.

CHAPTER - 2

INTRODUCTION TO COMPUTED TOMOGRAPHY

INTRODUCTION

Computed tomography (CT) originally known as computed axial tomography (CAT), is a medical imaging technique, which uses tomography for getting the internal details of an object from a large series of two-dimensional X-ray images taken around a single axis of rotation. However, nowadays, CT is not limited to the medical imaging only instead of it has spread over a wide application area. Now, CT is a powerful non-destructive evaluation (NDE) technique for producing two-dimensional or three-dimensional cross sectional images of an object from the flat X-ray images. Using CT, one can fetch the internal characteristics of an object such as dimension, shape, internal defects etc [25][26].

Even though, a CT image provides information not available in a traditional X-ray, this does not come without paying any cost. The radiation dose required for imaging a single CT slice is more than twice what it is required in general medical procedures. The effect of the radiation dose depends on various parameters. If a dose of large quantity is supplied within a very short period, biological effects may appear within hours and in the worst case it may cause death. Therefore, always some efforts are made to reduce the exposure time of X-rays to the patient. For reducing the exposure time a fast data capturing mechanism along with a fast processing system is required.

2.2 CALCULATION OF TOMOGRAPHIC IMAGES

Consider the case of an X-ray passing through a slice of tissue with four attenuation coefficients as shown in Figure 2.1. The output intensity I is given by

$$I = I_0 exp\left[-\left(\mu_1 + \mu_2 + \mu_3 + \mu_4\right)x\right] \tag{2.1}$$

Where, x is the distance traveled by the X-ray beam [26]. Rearranging Equation (2.1) and taking logarithm, we obtain the projection function as in Equation (2.2.d)

$$\frac{I}{I_0} = exp\left[-\left(\mu_1 + \mu_2 + \mu_3 + \mu_4\right)x\right] \tag{2.2.a}$$

$$ln\left(\frac{I_0}{I}\right) = (\mu_1 + \mu_2 + \mu_3 + \mu_4)x \qquad (2.2.b)$$

$$p(x) = ln\left(\frac{I_0}{I}\right) * x^{-1} = (\mu_1 + \mu_2 + \mu_3 + \mu_4) \qquad (2.2.c)$$

$$p(x) = \sum_{i=1}^{4} \mu_i \qquad (2.2.d)$$

It is clear from Equation (2.2.d) that a projection function is a simple summation of all different attenuation coefficients along a given path of X-ray beam.

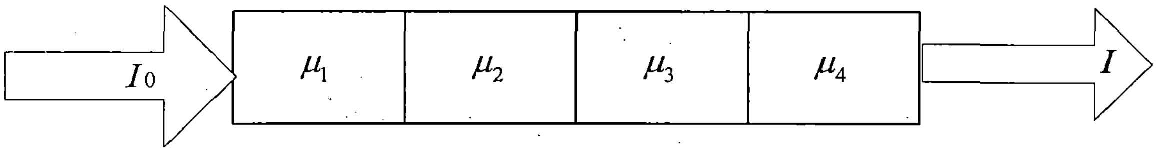


Figure 2.1: A single X-ray beam passing through a segment of tissue

In X-ray, CT contrast between different types of body tissues and bones is associated with the different attenuation coefficients of the material involved. The projection data collected over a range of angles forms the basis for tomographic image reconstruction. If the tissues in a cross section of object are assumed to have attenuation coefficients in a matrix form as in Figure 2.2, then for a particular row *i* the projection is defined as:

$$p_i(x) = \mu_{i_1} + \mu_{i_2} + \mu_{i_3} + \cdots + \mu_{i_m} \qquad (2.3)$$

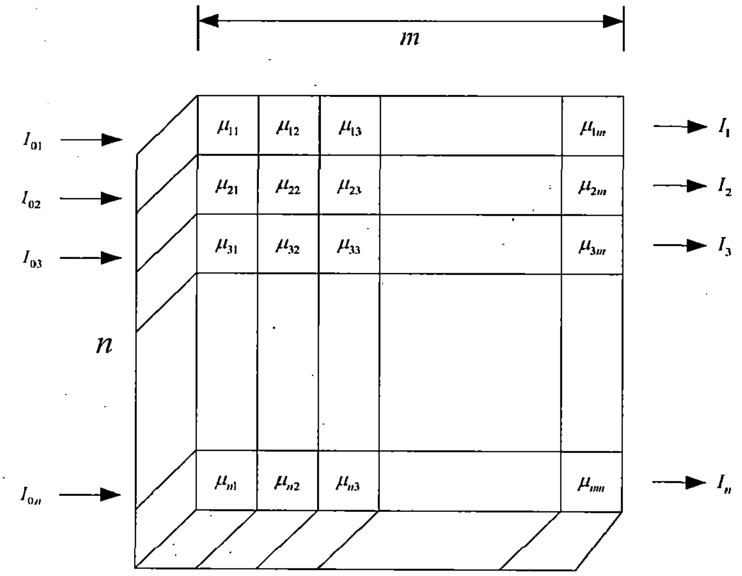

m
I_{01}
I_{02}
I_{03}
n
I_{0n}
μ_{11} μ_{12} μ_{13} μ_{1m}
μ_{21} μ_{22} μ_{23} μ_{2m}
μ_{31} μ_{32} μ_{33} μ_{3m}
μ_{n1} μ_{n2} μ_{n3} μ_{nm}
I_1
I_2
I_3
I_n

Figure 2.2: A 2-D matrix of attenuation coefficients representing a block of tissues

A scan shown in the Figure 2.2 is taken at zero degree with respect to horizontal. In a standard X-ray CT scanner, two types of motions are involved. Initially, the radiation source scans the object along a line for a given value of θ, and after the completion of the scan, the radiation source and detector array are rotated for a small angle $\Delta\theta$ and the linear scan is performed again for that view angle. This process is performed until all the projections over $180°$ rotation are collected. This is shown in Figure 2.3.

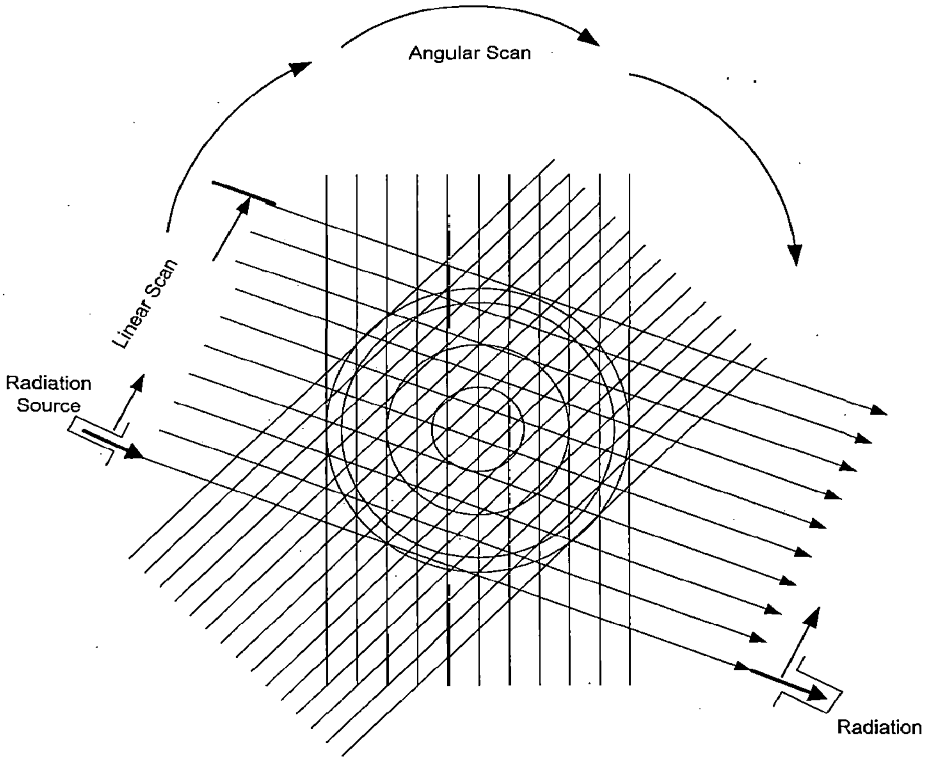


Figure 2.3: Parallel Beam and Angular Scanning

The goal of CT is determining the attenuation coefficient value of each type of tissues only based on these sets of projection functions. For final image reconstruction, each of these projection functions is backprojected on a blank image after some filtering.

The CT equipments used in Parallel beam tomography were first generation equipments of tomographic imaging. Now, these equipments have undergone various generations of development. The first generation was Hounsfield's configuration, and consisted of pencil beam source and detectors. Later, the CT equipment took the measurements by linear motion of source and detector for each projection. The next generation of CT introduced a small fan beam source and linear detector array. Projections were measured by translating and rotating the system around the object.

Third generation CT equipments were a slight improvement over the previous generation, as they used a wide fan bean source. The concept behind using a wide beam was to scan the whole object without any translation. The fourth generation CT equipments introduced a stationary circular detector array and a rotating wide fan beam source. Therefore, this generation CT equipments had only one rotating part that helped in reducing motion artifact largely. The fifth generation CT equipments are used for three-dimensional or volume imaging. These equipments use a cone beam source and a two-dimensional array of detectors. The mathematics involved in all these new generation CT equipments is derived from first generation parallel beam scanners. Therefore, the next section discusses the mathematics of first generation CT scanners.

2.3 PROJECTIONS

In the previous section, a term projection function was introduced. This projection function was nothing but a linear summation of all the different attenuation coefficients along a single beam of X-ray. In medical imaging also the word, projection has the similar meaning. Here, a projection is the collection of all detected radiations for a single linear scan at a given angle θ. This is illustrated with the help of Figure 2.4.

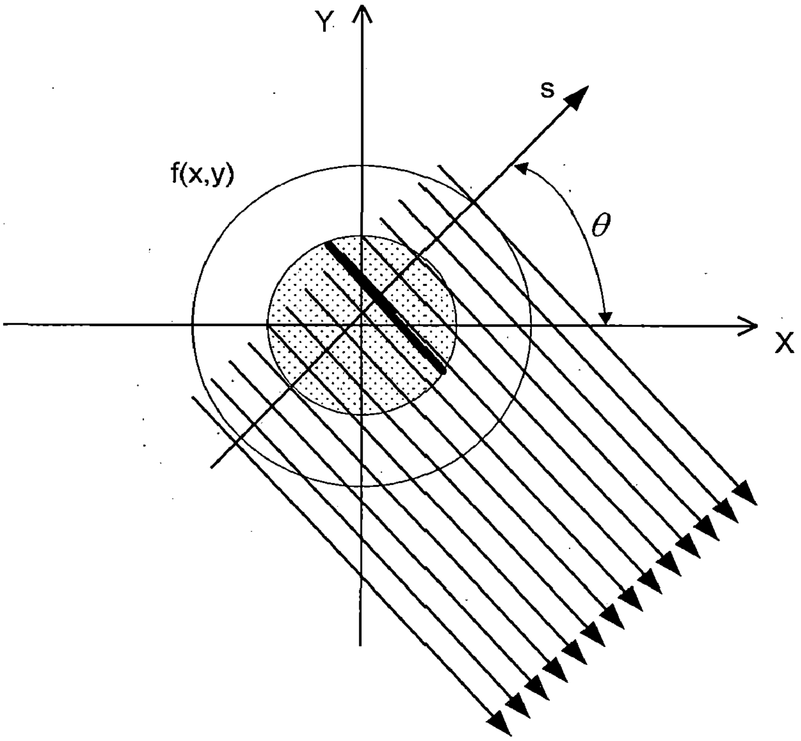


Figure 2.4: Parallel beam geometry and coordinate system

Here, *f(x,y)* represents the density distribution or attenuation coefficient distribution in a particular section of the object where, x and y are spatial coordinates. The figure shows a number of parallel lines passing through *f(x,y)* to represent a projection of the object taken at an angle θ. The symbol s is the perpendicular distance of particular line from the origin. Single line of projection is given in Figure 2.5.

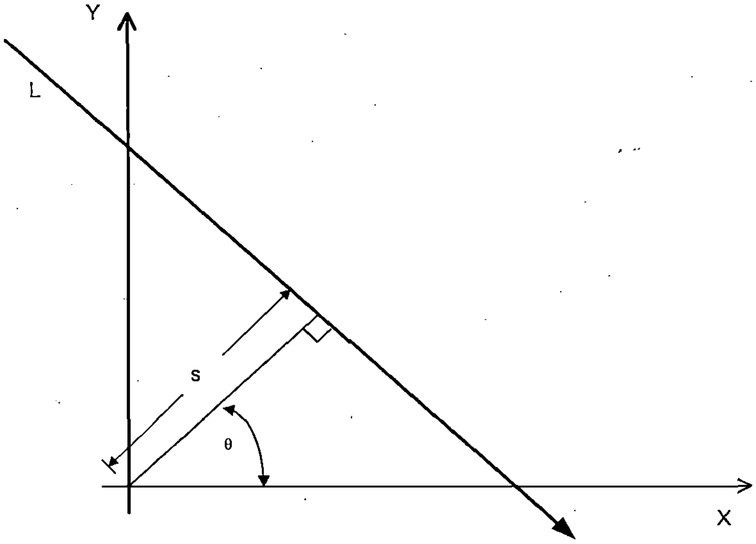


Figure 2.5: A Single beam from a parallel beam projection and its definition

From the Figure 2.5 the parametric equation of the given line L may be given as:

$$s = x\cos\theta + y\sin\theta \tag{2.4}$$

Let r be the distance of an element at (x,y) along the line L, then value of r can be calculated by using following relationship:

$$r = -x\sin\theta + y\cos\theta \tag{2.5}$$

Therefore, a new coordinate system is defined by rotating (x,y) by an angle θ in counterclockwise direction. The transformation from one coordinate system to other is found by Equations (2.6) and (2.7) given as:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} s \\ r \end{bmatrix} \tag{2.6}$$

And

$$\begin{bmatrix} s \\ r \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{2.7}$$

For line L given in Figure 2.5, the attenuation line sum can be calculated using following equation:

$$I_{\theta_L}(s) = I_0 exp\left[-\sum_r f(x,y)\Delta r\right] \tag{2.8}$$

$$= I_0 exp\left[-\sum_r f(scos\theta - rsin\theta, ssin\theta + rcos\theta)\Delta r\right] \tag{2.9}$$

Similarly, calculation of attenuation line sum for every line given in Figure 2.4 is carried out. These line sums for a given angle can be represented collectively by a single attenuation profile as:

$$p_\theta(s) = -ln\left[\frac{I_\theta(s)}{I_\theta}\right] = \sum_r (f(scos\theta - rsin\theta, ssin\theta + rcos\theta))\Delta r \tag{2.10}$$

If the attenuation coefficient is considered to be defined by a continuous function, then this formulation would become as:

$$p_\theta(s) = \int_{-\infty}^{\infty} f(scos\theta - rsin\theta, ssin\theta + rcos\theta)dr \tag{2.11}$$

The function $p_\theta(s)$ is called the projection of the function $f(x,y)$ along the angle θ. Every projection contains some information about the object. However, for a faithful reconstruction of the original object many of these projections are required. Usually these projections are taken for various values of θ over the range 0 to 180°. A projection is a one-dimensional vector of received intensities in (s, θ) space. All projections are stacked together to form a two-dimensional data set $p(s,\theta)$, which is called a sinogram. This transformation of function $f(x,y)$ in to the sinogram $p(s,\theta)$ is called the Radon transform (RT).

2.4 THE RADON TRANSFORM

The two-dimensional RT is an operator that maps a real function $f(x,y)$ into a set of its line integrals. Mathematically, RT is given by the following equation [1]-[4]:

$$\Re\{f(x,y)\} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\delta(x\cos\theta + y\sin\theta - s)dxdy \tag{2.12}$$

where, $\Re$ is the Radon operator and δ is the Dirac delta function. The term $(x\cos\theta + y\sin\theta - s)$ is the parametric equation of the line as shown in Figure 2.5. The RT maps a two-dimensional function in rectangular space (x, y) to radon space characterized by (s, θ). Physical significance of radon space coordinates (s, θ) can be understood by taking an illustration given in Figure 2.4. The values of s are obtained by linear translation of source and detector array and the values of θ are obtained by rotating the source-detector pair around the body of the object.

Theoretically, Radon transform is defined over an infinite space. However, in a practical situation the object represents a limited space. In addition to this, the coordinates of image space (x, y) assume a set of discrete values only. Therefore, the resultant radon space would be a discrete space and can be described with the help of summations instead of integrations. A detailed discussion of Discrete Radon transform (DRT) [12] [13] is given in following section.

2.5 DISCRETE RADON TRANSFROM

The Radon transform of the object consists of continuous projections for an infinite number of angles in a given range. However, practically there are only a number of discrete projections taken at finite discrete intervals over a given range of angles. Projections are discrete because of another practical limitation that is the detectors are discrete recording elements. Each detector records the intensities independent of neighboring detectors. Therefore, the practical Radon transform is not a continuous transformation. A discrete formulation of the practical RT is given in Equation (2.13) as:

$$p(s,\theta)=\sum_{L(s,\theta)} f(x,y)\Delta r \tag{2.13}$$

where, L(s, θ) represents set of lines passing through the object *f(x,y)*. Here, θ assumes a value from finite discrete values. Literature reveals several approaches to compute DRT. These methods differ mainly in the way of calculating the weight factor Δr, for a given sampled input image. The summation and sampling terms requires some interpolation. There are various interpolation methods in use such as nearest neighbor interpolation, linear interpolation, line length and pixel area [10]. For a hardware implementation, different interpolation schemes offer a different computational loading. From the literature and initial MATLAB simulations, it linear interpolation was found to give the best compromise between the qualities of reconstructed image and the computational loading [10].

2.6 CONCLUSION

Parallel beam tomography forms the mathematical basis of all other generations of CT. All mathematical relations for later generations can be derived from parallel beam tomography. Therefore, a digital implementation of Radon transform for the former can be altered appropriately to get an implementation for any later generation CT. Since, RT is not directly invertible to image space; DRT becomes a better choice for implementation. Sampling and summation need some interpolation. Linear interpolation technique is a better choice as it offers a good trade off between quality of result and complexity of calculation.

CHAPTER - 3

IMAGE RECONSTRUCTION USING BACKPROJECTION

3.1 INTRODUCTION

Reconstruction of original two-dimensional function $f(x,y)$ is an important problem in the field of non-invasive imaging. Image reconstruction is the process of estimating an image $f(x,y)$ from a set of one-dimensional projection. We cannot reconstruct a two-dimensional function with the help of a single projection due to loss of information during the projection operation. However, if a number of projections taken at different angles are available, one can estimate the information contained in the original object. The required number of projections for a given image is still a topic of debate. However, an estimate about the required number of projections was given by Ronald Bracewell [6].

The problem of image reconstruction is constrained by many parameters. First important constraint is the number of projections available for image reconstruction. Another constraint is the view of projection, as it may be a full view projection or a truncated projection. In case of truncated projections, extrapolation techniques are used to complete the projection before reconstructing the image [17].

In practice, there are many ways to reconstruct a two-dimensional function from projection data. In this chapter, before discussing the image reconstruction techniques, a very basic theorem relating the Radon space with Fourier space is described. A brief description of most of the important reconstruction methods is given in a later section. Finally, the method implemented during the present thesis work is discussed in detail along with comparisons with other methods.

3.2 THE FOURIER SLICE THEOREM

The Fourier slice theorem builds the fundamental basis for a number of image reconstruction techniques. It relates the one-dimensional Fourier transform of a projection of a two-dimensional function $f(x,y)$ with the two-dimensional Fourier transform of $f(x,y)$. Let $F(u,v)$ be the Fourier transform of the image $f(x,y)$. Mathematically, it is given by Equation (3.1).

$$F(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)e^{-j2\pi(ux+vy)}dxdy \tag{3.1}$$

Also, let $S_\theta(\omega)$ be the Fourier transform of the projection $p_\theta(s)$, i.e.

$$S_\theta(\omega) = \int_{-\infty}^{\infty} p_\theta(s) e^{-j2\pi\omega s} ds \tag{3.2}$$

The Fourier slice theorem is stated as, "The one-dimensional Fourier transform of a projection $p_\theta(s)$, taken with respect to *s*, is equal to the central slice, at angle *θ*, of two-dimensional Fourier transform of the object *f(x,y)*" [1] [3]

i.e. if

$$f(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(u,v) e^{j2\pi(ux+vy)} dudv \tag{3.3.a}$$

Then,

$$S_\theta(\omega) \equiv F(\omega,\theta) \tag{3.3.b}$$

Where, *F(ω,θ)* denotes the value of *F(u,v)* along a line at an angle *θ* with the *u*-axis as shown in Figure 3.1.

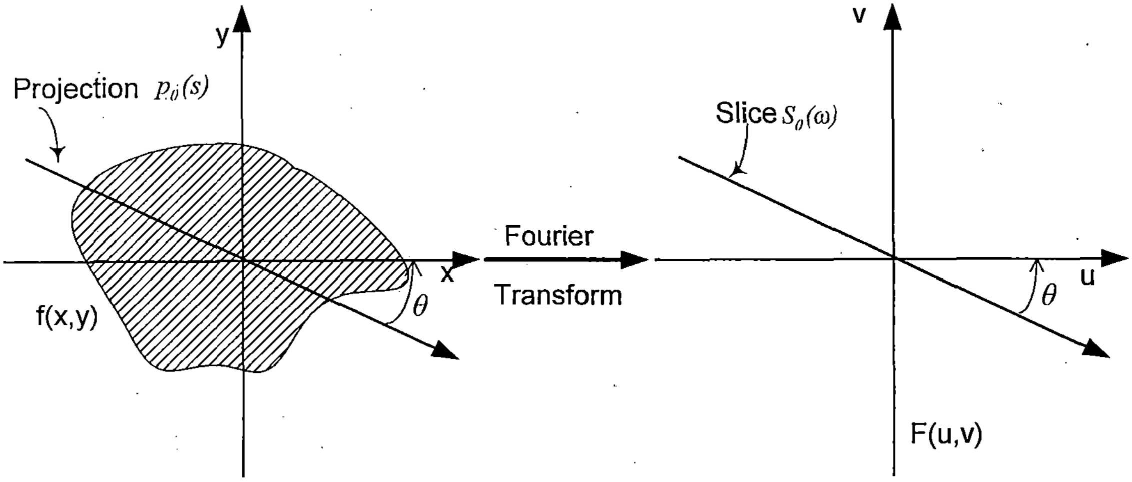


Figure 3.1: A simple example of Fourier slice theorem

Therefore, Fourier slice theorem indicates that by taking the projections of the object at angles $\theta_1, \theta_2, \theta_3, \ldots, \theta_k$, and taking Fourier transform of them, one can determine the values of *F(u,v)* on the radial lines. If an infinite number of projections were taken, then *F(u,v)* would be known at all points in uv-plane. Knowing *F(u,v)*, the image *f(x,y)* can be recovered by taking the two-dimensional inverse Fourier transform, as given by Equation (3.4).

$$f(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(u,v)e^{j2\pi(ux+vy)}dudv \qquad (3.4)$$

In most of the practical cases, such as, image two-dimensional function is a bounded function in space. Suppose, if the function *f(x,y)* is bounded by $-d/2 < x < d/2$ and $-d/2 < y < d/2$, then for the purpose of computation, Equation (3.4) must be modified into a two-dimensional discrete-time Fourier transform (2-D DTFT).

Practically, it is impossible to take infinite number of projections over a finite space due to computational limitations. Therefore, if there are N numbers of projections available, frequency domain coordinates will vary from $-N/2$ to $N/2$ only. It is obvious from Figure 3.1 that if the number of projections is finite, then the density of values at higher frequencies will be sparse and it will be very dense at the low frequencies. If an inverse of this function is taken, there will be excessive blurring of the resultant image. Therefore, some kind of preprocessing is desired before taking the inverse transform as to get a faithful reconstruction of the original object. The knowledge of Fourier slice theorem enables us to correlate the Radon space with the Fourier space. Some important reconstruction techniques are discussed in the following section.

3.3 IMAGE RECONSTRUCTION TECHNIQUES

There are several algorithms existing to reconstruct an image from projections. *Following are the few important methods used for this purpose:*

- Brute force inversion technique
- Iterative reconstruction technique
- Fourier-based technique
- Filtered backprojection technique

Brute force and iterative techniques are the oldest techniques used for this purpose. However, introduction of "Fourier slice theorem" revolutionized the field of image reconstruction. Fourier based methods were accepted widely because of better understanding and prior knowledge about Fourier transforms. Another idea was to take the direct inverse of Radon transform, which gave rise to Backprojection technique. Filtered backprojection technique is a complete direct inversion technique. This method has received a good acceptance and it is being used in most of the modern medical image processing machines.

3.3.1 Brute force inversion technique

This inversion technique takes help of linear algebra in estimating the pixel values in the image to be reconstructed. It formulates a set of simultaneous linear equation from the given projection data. Any algorithm used for solving simultaneous linear equations can solve this set of simultaneous liner equations. Consider, the following example having six projections over a 2x2 matrix.

$$\begin{array}{c} \begin{array}{l} \\ \end{array} \begin{bmatrix} a & \mathrm{b} \\ c & \mathrm{d} \end{bmatrix} \begin{array}{l} 10 \\ 8 \end{array} \\ 6 \quad 10 \quad 8 \quad 12 \end{array} \qquad (3.5.a)$$

These can be organized in a matrix form, which represents a set of six simultaneous equations as follows:

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} 10 \\ 8 \\ 10 \\ 8 \\ 12 \\ 6 \end{bmatrix} \qquad (3.5.b)$$

The set of simultaneous equations shown in Equation (3.5.b) represents a simple problem in linear algebra. This is again an over-determined problem because there are six equation and only four unknowns. However, this technique may require to solve hundreds of linear equation for a single slice. Therefore, it is not suitable for real-time implementation.

3.3.2 Iterative reconstruction technique

Iterative reconstruction technique was first technique to be used in practical tomographic image reconstruction. Basic mathematics involved in this technique was algebra. Therefore, it was also known by algebraic reconstruction technique (ART). This technique follows a stepwise procedure to compute the estimates for the image pixels. The steps involved are given as follows:

1. Make an initial guess of the solution
2. Compute the projections based on this guess
3. Improve the initial guess by comparing the results obtained from previous step with the actual projection values. The improvement is made using the following relationship

$$P^{i+I} = P^{i} + g(actual - computed) \qquad (3.6)$$

Here, P^i denotes one-dimensional array of all pixels at i^{th} iteration and g is a multiplication coefficient, which is a real number in the range [0.0, 1.0]. The multiplication coefficient is used to control the rate of improvement. Selection of a suitable value is done in an inverse relation with the number of projections available. For a large number of projections its value is taken to be sufficiently small and vice versa.

The ART was used practically in initial CT scanner to reconstruct the image. This technique has certain drawbacks such as, slow convergence and high sensitivity to noise. With more noise, convergence slows down further and may lead to an incompatible result.

3.3.3 Fourier-based technique

The principle of Fourier based image reconstruction is to take one-dimensional Fourier transform of the projection data and fill the two-dimensional Fourier space using Fourier slice theorem. Suppose, the image $f(x,y)$ is bounded by $-d/2<x<d/2$ and $-d/2<y<d/2$, then we can reconstruct the image by using Equation (3.7) given as:

$$f(x,y)=\frac{1}{d^2}\sum_m\sum_n F(\tfrac{m}{d},\tfrac{n}{d})e^{j2\pi(\frac{m}{d}x+\frac{n}{d}y)} \qquad (3.7)$$

Here, variables m and n will vary from –N/2 to N/2 for N number of projections. This equation can be implemented rapidly using the Fast Fourier transform (FFT) algorithm provided the N^2 Fourier coefficients are known. The concept of Fourier-based reconstruction is shown with the help of a block diagram in Figure 3.2.

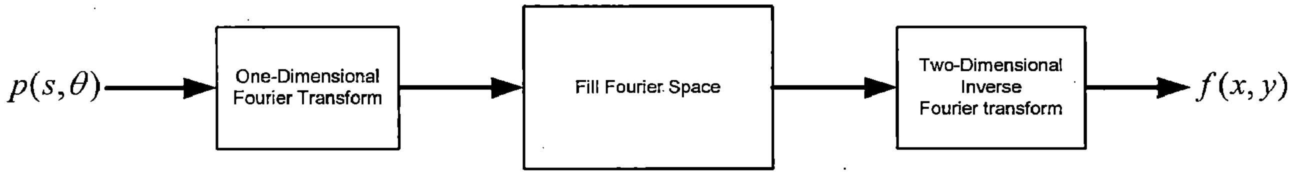

$p(s,\theta)$
One-Dimensional Fourier Transform
Fill Fourier Space
Two-Dimensional Inverse Fourier transform
$f(x,y)$

Figure 3.2: A conceptual block diagram of Fourier-based Reconstruction

Since, we have a limited number of projections, the function $F(u,v)$ is known only along a finite number of radial lines. In order to be able to use Equation (3.7) one must have to interpolate from these radial lines to the points on a square grid. Theoretically, one can calculate all N^2 coefficients required in Equation (3.7) provided as many values on these radial lines are given. However, this interpolation is computationally intensive and often leads to unstable results. Moreover, interpolation error at higher frequencies is much higher in comparison to that at lower frequencies, this result in degradation of the image quality.

3.3.4 Filtered backprojection technique

In backprojection, the measurement obtained at each projection is projected back along the same lines and at the same angles. Thus, each projection is smeared back on a blank square grid and then the contribution of each projection is accumulated for each pixel. The estimated gray level for each pixel is then calculated by dividing the corresponding accumulated value by total number of projections. The process of backprojection is shown with a simple example in Figure 3.3.

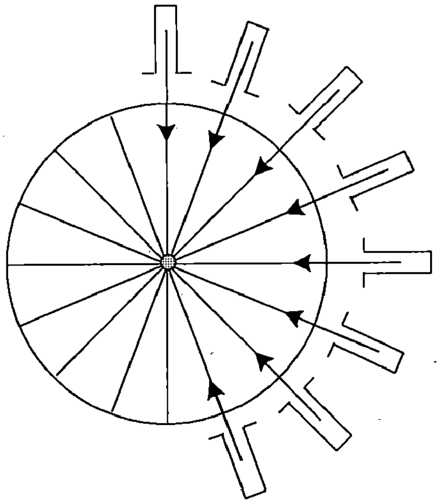


Figure 3.3: Backprojection of a point mass on origin

Backprojection algorithm is based on use of a backprojection operator. A backprojection operator $\mathfrak{B}$ is defined as:

$$b_\theta(x,y) = \mathfrak{B}\{p_\theta(s)\} = \int p_\theta(s)\delta(x\cos\theta + y\sin\theta - s)ds \tag{3.8}$$

The quantity $b_\theta(x,y)$ is the backprojection density due to projection $p_\theta(s)$. Backprojection operator assigns the value of projection to the entire pixels that are falling on the line $x\cos\theta + y\sin\theta - s$. Adding up the projections at all angles, we obtain

$$f_b(x,y) = \int_0^\pi b_\theta(x,y)d\theta \tag{3.9.a}$$

$$= \int_0^\pi \int_{-\infty}^{\infty} p_\theta(s)\delta(x\cos\theta + y\sin\theta - s)dsd\theta \tag{3.9.b}$$

The result obtained from Equation (3.9.b) will give a distorted estimate of the original function $f(x,y)$. For practical purposes, this distorted reconstruction is not acceptable. It is desirable to reduce this blur without using any two-dimensional transform. The Fourier slice theorem provides such possibilities. With the help of Fourier slice theorem Equation (3.9.b) can be re-written as given in Equation (3.10)

$$f_b(x,y) = \int_0^{\pi}\int_{-\infty}^{\infty}\left[\int_{-\infty}^{\infty} S_\theta(\omega)e^{j2\pi\omega s}d\omega\right]\delta(x\cos\theta + y\sin\theta - s)dsd\theta \qquad (3.10)$$

Integrating with respect to s and replacing $S_\theta(\omega)$ with $F(\omega,\theta)$, above equation becomes

$$f_b(x,y) = \int_0^{\pi}\int_{-\infty}^{\infty} F(\omega,\theta)e^{j2\pi\omega(x\cos\theta + y\sin\theta)}d\omega d\theta \qquad (3.11)$$

The two-dimensional inverse Fourier transform in polar form is defined as:

$$f(x,y) = \int_0^{2\pi}\int_0^{\infty} F(\omega,\theta)e^{j2\pi(x\cos\theta + y\sin\theta)}\omega d\omega d\theta \qquad (3.12)$$

This can also be written as

$$f(x,y) = \int_0^{\pi}\int_{-\infty}^{\infty} F(\omega,\theta)e^{j2\pi(x\cos\theta + y\sin\theta)}|\omega| d\omega d\theta \qquad (3.13)$$

It is necessary to use $|\omega|$ since, the integration includes the negative values. Comparing Equations (3.11) and (3.13) gives that the original function and the reconstructed function differs only by a weighting term $|\omega|$. This term accounts for the blurred reconstruction of $f(x,y)$. Therefore, for fixing this problem the projections are pre-weighted with this weighing term $|\omega|$ using filtering before backprojection.

This reconstruction technique is called filtered backprojection (FBP) technique. Most of the modern day's tomographic imaging systems use this technique for image reconstruction purpose. The key advantage of this technique is that it involves only one-dimensional Fourier transform for filtering purpose.

Digital computer implementation of FBP would require a discrete representation of Equation (3.13). Continuous to discrete space transformation is a straightforward process. The integration becomes summation and the derivative term happens to be the finite differences. The only problem in this conversion is the weighing term $|\omega|$. This term clearly represents a ramp filter. A ramp filter is not a stable system. Instability of this filter makes it useless in practice. However, an approximation can be made because the input data can be assumed to be band limited. This is true in practice, because of the sampling involved in measurement process. Ramachandran and Lakshminarayanan [18]

[19] gave a suitable approximation of the above-mentioned frequency domain filter, which is popularly known as Ram-Lak filter. Impulse response of the Ram-Lak filter is given as:

$$h(nT) = \begin{cases} \frac{1}{4T^2} & n = 0 \\ 0 & n \text{ even} \\ -\frac{1}{n^2 p^2 T^2} & n \text{ odd} \end{cases} \tag{3.14}$$

There are many other approximations of ideal ramp filter [2]. These approximations provide different trade-offs between spatial frequency resolution and the noise immunity.

3.4 CONCLUSION

Iterative reconstruction technique is a practically working technique. The drawback of this technique is that it is slow in convergence and very much sensitive to noise. In presence of excessive noise, it may not converge at all. Fourier-based reconstruction takes the advantage of Fourier slice theorem to find a rigorous mathematical solution of image reconstruction problem. However, due to the complexities involved with interpolation in frequency domain and two-dimensional transforms, it is also not suitable for high-speed implementation. Filtered backprojection technique shows the best promising features for digital implementation, as it needs only one-dimensional transforms. Also, filtering can be performed in spatial domain itself by designing a FIR filter from the given impulse response. It is also clear that the image reconstruction algorithms are inherently serial in nature. For a parallel implementation, a large amount of on-chip or off-chip memory to handle intermediate results will be required.

CHAPTER - 4

PROPOSED ALGORITHM FOR DIGITAL IMPLEMENTATION OF IMAGE RECONSTRUCTION

INTRODUCTION

The architecture used in the implementation of Filtered backprojection algorithm is presented here. Image reconstruction operation needs to take inverse of Radon transformed data, which Is available in the form of projections. Therefore it is useful to go through a brief discussion about implementation of Radon transform, before Backprojection algorithm. It has been found that both operations are very much similar in terms of operators required and mathematically backprojection is a direct inverse of Radon transform.

Proposed algorithms for implementations are formulated considering various constraints. These constraints are decided by the inherent serial nature of these transforms as well as the resources available for their implementation. In current work, the design is purely serial in nature. However, parallel implementation of the algorithm is possible, but is not feasible to implement it with the available resources with current work. The main constraint is imposed by the memory required to store the intermediate results. An image processing application need a bulk amount of data to be processed and hence requires a bulk amount of on-chip as well as off-chip memory. In support of the current work, asynchronous memory resources are available off-chip. However, it is required to design dedicated memory interfaces to use these memories, which limits the performance of the circuit largely. Memory resources available on-chip is used to fulfill the need of a fast intermediate storage.

Input digital image is a two-dimensional array of pixel, where each pixel is represented by 8-bit data value. In other words, input image has 256-color resolution. The calculation of forward and inverse transform involves complex trigonometric functions. Therefore, integer arithmetic introduces large rounding errors. A floating-point representation can be used to address this problem. A floating-point representation gives best performance in terms of accuracy. However, it needs a large amount of logic and routing resources on the chip and larger memory for storing the intermediate results.

A fixed-point representation provides very good trade-off between the accuracy and requirement of hardware [25]. Since in the current work the ultimate goal is to restore the visual qualities of the image, a very good extent of accuracy is not necessary. Therefore, in the current work a fixed-point representation is used to optimize the speed and resource utilization. In this chapter, first and second sections describe the development of algorithm for Radon transform and its implementation on FPGA. Then in the next section algorithm for implementation of Backprojection is formulated. Finally, this chapter is concluded with discussion of implementation of Filtered backprojection algorithm on FPGA.

4.2 PROPOSED ALGORITHM FOR RADON TRANSFORM IMPLEMENTATION

The calculation of Radon transform is generally not required. The data obtained from CT scanner is already in the Radon transform form. Calculation of Radon transform is required in some special cases where a three-dimensional image has to be constructed from two-dimensional slices. The implementation starts with the initialization of the Radon space *p(s, θ)* with a zero value. The input image *i(x, y)* contains the data to be transformed. RT is calculated one projection at a time. For the calculation of each projection, all pixels in the input image are passed once through the processor element. Each pixel from the input image is passed in a raster scan way and the corresponding address in radon space is calculates as:

$$s = xcos\theta + ysin\theta \qquad (4.1)$$

However, one can see that the value *s* calculated in above equation is not an integer number in most of the cases. Therefore, truncation is used to get a valid address in Radon space. For truncation, the lower bound of the real number represented by *s* is taken as:

$$bin = floor(s) \qquad (4.2)$$

This truncation introduces an error. This error can be compensated by using the remaining fractional value for interpolation. Therefore, the remaining fraction value is stored in a different variable.

$$fract = s - bin \qquad (4.3)$$

The origin for an *MxM* image is taken at (*M*/2, *M*/2). Although this convention is easily realizable in software (as used by MATHWORKS), in a digital hardware implementation it is not so straightforward. Since origin has been chosen in the center of the image, at least one of the image coordinates (*x*, *y*) assumes negative values in three quadrants.

The first problem arising from this situation is to generate the values of x and y in the range $-M/2$ to $M/2$. Moreover, the result of Equation (4.1) gives some negative values of s. Since an address in the Radon space can have a negative value, s is shifted properly to give a valid address. Therefore, the first challenge for a complete on-chip implementation is to ensure a correct address generation. Figure 4.1 shows a simple example of address calculation scheme for Radon transform.

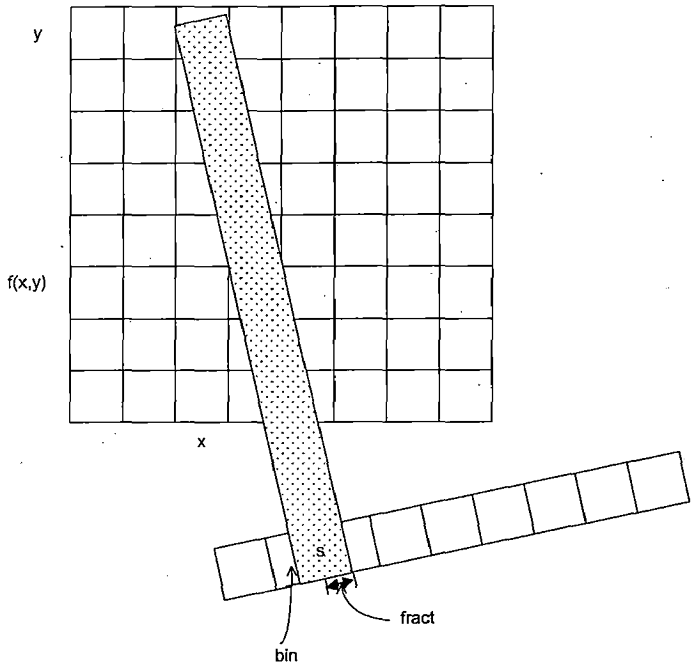


Figure 4.1: Address calculation for Radon space

If the address generation process has been implemented successfully, the only work left is data processing. Data processing is a relatively straightforward task, as it requires interpolation and addition only. The flow of whole algorithm is demonstrated in Figure 4.2. Here $\hat{p}(bin,\theta)$ denotes the modified value of the Radon space element $p(bin,\theta)$ after each pass of input image. Once the execution is complete, $p(bin,\theta)$ contains the radon transform of the input image.

4.3 PROPOSED STRATEGY FOR RADON TRANSFORM IMPLEMENTATION

The key goal is to get a high-speed implementation of RT for an image. The fast RT calculation is not required in general, but in few cases such as iterative reconstruction algorithms, where a fast RT capability is desirable.

This section is dedicated to discuss the strategy used for the digital implementation of the algorithm presented in pervious section. In a digital hardware every variable in the given algorithm has to be modeled as a signal or register. A signal carries the information from one point to another point in the architecture where as, a register stores the information for later use. A registers assigns the stored value to a signal for further use at a predefined moment of time. Here every signal is represented by an array of bits. An optimal number of bits are taken for each signal to minimize the hardware utilization, which in turn minimizes the cost of the hardware.

```
Input : Image $i(x,y)$;
Initialization : $p(s,\theta)$ with zero;
set $\theta = 0$;
while($\theta < 256$)
    Repeat :  for each pixel in input image, do
              $s = x\cos\theta + y\sin\theta$;
              $bin = \text{floor}(s)$;
              $fract = s - bin$;
              $\hat{p}(bin,\theta) = p(bin,\theta) + i(x,y) * (1 - fract)$;
              $\hat{p}(bin+1,\theta) = p(bin+1,\theta) + i(x,y) * fract$;
              $\theta{+}{+}$;
end while;
```

Figure 4.2: Algorithm for implementation of Radon transform

A block diagram of the circuit designed to compute the Radon transform is given in Figure 4.3. Here input image and output radon data are stored in different external memories. SRAM1 Interface block is designed to interface the input SRAM with the custom circuit. This circuit is responsible for address generation for SRAM1 and for generating the required control signals to communicate with the external memory. This interface uses a seven state finite-state-machine (FSM) to generate the control signals in a proper sequence. However, as this circuit is generating the address locally, a control of address generation is also necessary. Rest of the circuit can accept a new data value only when the current data value has been processed completely. Therefore, a handshaking signal between the output interface and input interface circuits has been used to address this difficulty. A sophisticated variable wait states insertion technique is used with the help of this handshaking signal. In Figure 4.3, signal trig3_int generated by SRAM2 Interface circuit shows represents that signal. Initially, the FSM of SRAM1 Interface goes through all the previous states and then waits for a variable number of cycles in a wait state until signal trig3_int becomes high. Number of variable wait cycles depends on the number of the sates in the FSM running in SRAM2 Interface circuit. In terminologies of digital design, this is a case of interacting state machine design. Therefore, after going through a complete cycle, FSM of SRAM1 Interface sends a signal to the local address generation block to increment the address. SRAM1 Interface circuit read a single pixel from the input image in every cycle and sends it directly to the SRAM2 Interface circuit. In SRAM2 Interface circuit, a process is designed to works as a Radon processing element. This process takes pixel data coming from SRAM1 Interface and *fract* coming from Radon address generator block, shown in Figure 4.3. This process runs in concurrency with the FSM of SRAM2 Interface circuit. Depending on the states of FSM, this process performs different arithmetic operations on the data given in the Figure 4.2.

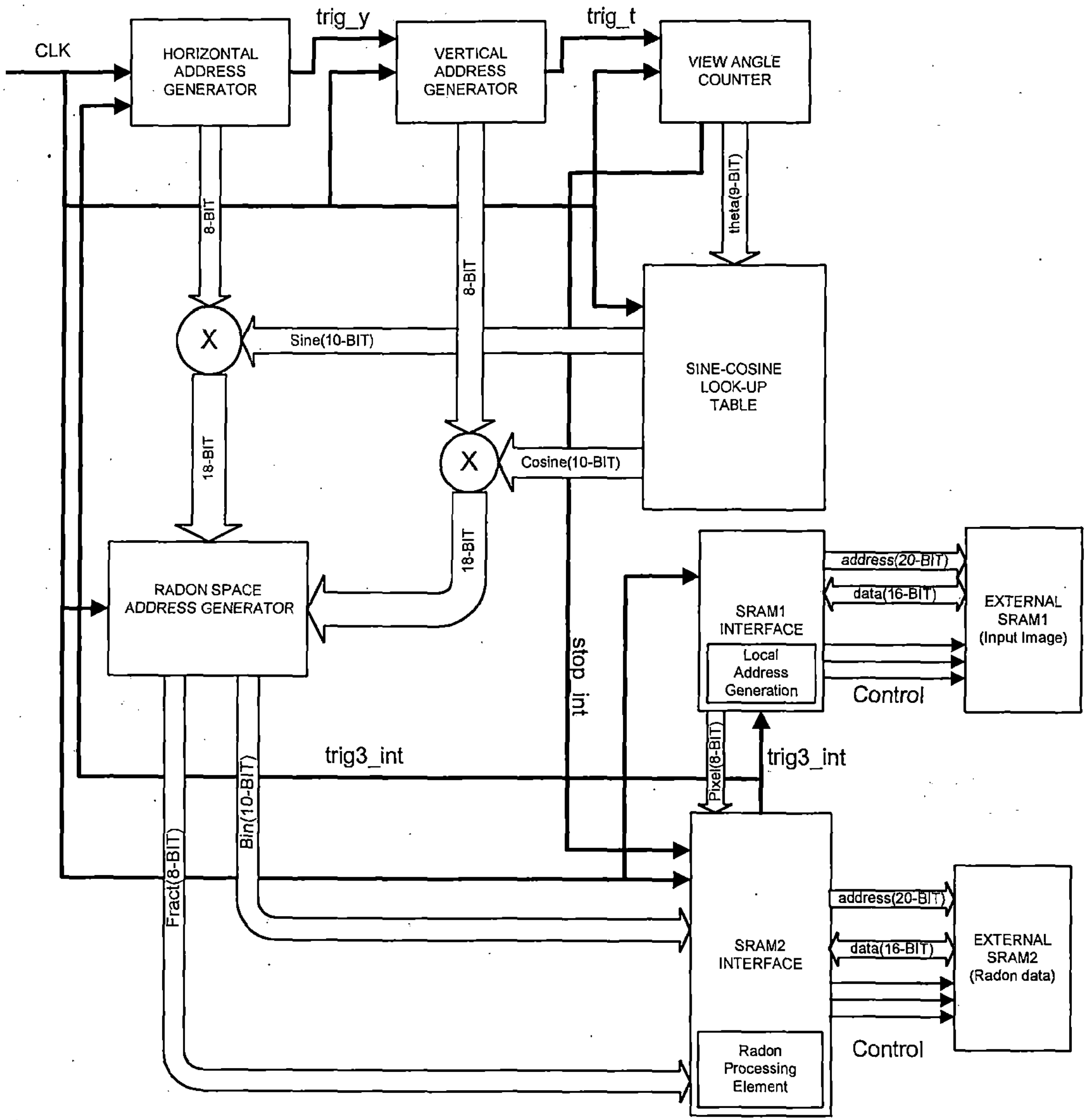


Figure 4.3: Schematic diagram for Radon Transform computation circuit

In this design, another important module is SRAM2 Interface. This circuit connects the custom circuit with another external SRAM, which is being used to store the radon data. As it can be seen that for every input pixel, two consecutive values of Radon data is being modified. For this, data is to be read from external SRAM, modified and ultimately written back to the SRAM, twice for every input pixel. This becomes a complicated memory access process, known as *Read before Write* memory access. This type of memory access should be addressed properly to ensure that the data written back after processing is a valid one. In the current work, SRAM2 Interface circuit uses a twenty-five state FSM to solve this problem. At the end of every a single input pixel has been processed completely and related results are saved on external SRAM. In addition to this, SRAM2 Interface circuit also sends a handshaking signal trig3_int to SRAM1 Interface circuit as well as to the horizontal address generator circuit.

The third important part of the design is a collection of modules that generate the address for radon space. Calculation of radon space address involves trigonometric functions. Because of the complexities involved in designing a customized circuit for these trigonometric functions, most of the earlier custom designs perform this operation off-chip and then load these values in another input memory [10]. This needs a large amount of off chip memory and further slows down the speed of design because of low speed interfaces. In the current work, a parameterized Xilinx IP core is used for this purpose. This IP core is instantiated in form of a read-only-memory (ROM), which stores the values of sine and cosine over a full wave. This parameterized core provides many options regarding the width of input *theta*, output precision, and signed/unsigned representation of output. This ROM works as a look-up table. For a particular value of input *theta,* it generates one value of sine and one value of cosine. The output of this table is directly usable in our design to perform the calculation of variable *s.* The Radon-space address generator module performs the operation of calculating s and its truncation to generate signals *bin* and *fract.* Horizontal address generator and Vertical address generator circuits generate the values of *x* and *y* coordinates in a raster scan fashion. The horizontal address generator module sends a trigger signal *trig_y* to vertical address generator module at the end of each row of the input image. Similarly, the vertical address generator sends a trigger signal *trig_t* for View angle counter module. A positive pulse on signal *trig_t* indicates that the input image has been passed completely for that particular view angle. Therefore, View angel counts up to next value of *theta* at each *trig_t* pulse. This design is a complete custom circuit for calculation of RT. It only requires input image in its input memory and it provides output after performing computation in output memory.

4.4 PROPOSED ALGORITHM FOR FILTERED BACKPROJECTION

Backprojection is a direct inverse of Radon transform. In previous discussion, we found that measurement of a projection is nothing but summation of pixels falling on the parallel lines along the direction of projection. However, backprojection operation is smearing back that projection along the same line on the image grid, and assigning its values to all the pixels falling along those lines. There are two possible ways for the implementation of backprojection algorithm. In first, the coordinates of image space are calculated with the help of radon-space coordinates, and the value of the corresponding projection is accumulated in all pixels addressed in this way. This method involves truncation in the values of both image coordinates, which results in huge rounding errors and hence it is rarely used. In the second method, the radon-space coordinates are calculated in the same way as shown in Figure 4.2. For a particular angle, each pixel of image is updated by a radon data value, which is selected by the coordinates of that particular pixel. This method requires truncation in a single variable and it can be easily compensated by using interpolation.

The implementation of backprojection starts with the initialization of the output image i(*x*, *y*) with a zero value. Each projection is projected back and its value accumulated over the image grid in a sequence. The algorithm presented here reads one pixel of from the output image at a time and upgrades its value for the current projection. In this way, all pixels of the output image are passed through the circuit once for every projection.

The generations of radon space coordinate *s* and discrete address *bin* and interpolation factor *fract* are done exactly as in the case of computation of Radon transform. Therefore, Equations (4.1), (4.2), and (4.3) are used again in this algorithm for the purpose of address generation. It is clear from Figure that during backprojection, a strip of unit width may not pass through a pixel completely. Mathematically this means that in a projection $p_\theta(s)$, a pixel does not contribute completely to a single point in the radon space. A single point in radon space does not completely correspond to a pixel in the image space. Therefore, an interpolation factor *fract* is used to derive the exact contribution for a particular pixel. The flow of the operations during backprojection is given in the Figure 4.4.

```
Input: Radon transfrom $p(s,\theta)$;
Initialize: image $i(x,y)$ with zero;
set $\theta = 0$;
while ($\theta < 256$)
        Repeat: For each pixel in input image, do
                $s = x\cos\theta + y\sin\theta$
                $bin$ = floor($s$);
                $fract = s - bin$;
                $E = p(s,\theta) * (1 - fract) + p(s+1,\theta) * fract$
                $\hat{i}(x,y) = i(x,y) + E$;
                $\theta++$;
end while;
```

Figure 4.4: Algorithm for implementation of backprojection

In Figure 4.4, E is an intermediate variable that is added with the current value of $i(x,y)$ to give the updated value $\hat{i}(x,y)$. After the backprojection operation is over, the accumulated data needs to be divided by the number of projections used in order to get the gray value for that pixel. This image is a blurred reconstruction of the original image. Generally, this image is hopelessly blurred and not acceptable for application purpose. Form previous discussion it has been found that the projection data must be filtered before backprojection. The required filter impulse response is given in previous chapter. Therefore, the Radon transform data is filtered and then backprojected using the propose algorithm.

4.5 IMPLEMENTATION STRATEGY FOR PROPOSED FILTERED BACKPROJECTION ALGORITHM

The implementation of backprojection algorithm given in Figure (4.4) is quite similar to that of the Radon transform case, discussed in previous section 4.2. The goal of backprojection is to obtain a reconstruction of the original image. However, mathematically it is known that this reconstruction will not be in a usable form because of too much blurring. This blurring effect has to be reduced by pre-filtering the projection before backprojection. Further, filtering can be performed either in space or in frequency domain. In the present work, a Finite Impulse Response (FIR) filter has been designed to perform the filtering in space. Each one-dimensional projection is filtered and then backprojected in order to get a faithful reconstruction of the original image. Filtering and backprojection operations can be performed in two separate steps. Here, proposed design uses partial parallelism in both processes. As it is obvious from the discussion of filtered backprojection operation that backprojection is performed by taking one projection at a time. This fact gives an opportunity to start backprojection operation as soon as the first projection has been filtered completely.

In the current discussion, first implementation of the backprojection operation has been discussed separately and then a proposed scheme for Filtered backprojection operation is given. The block diagram shown in Figure 4.5 represents the organization of various hardware modules required in the design of a backprojection circuit.

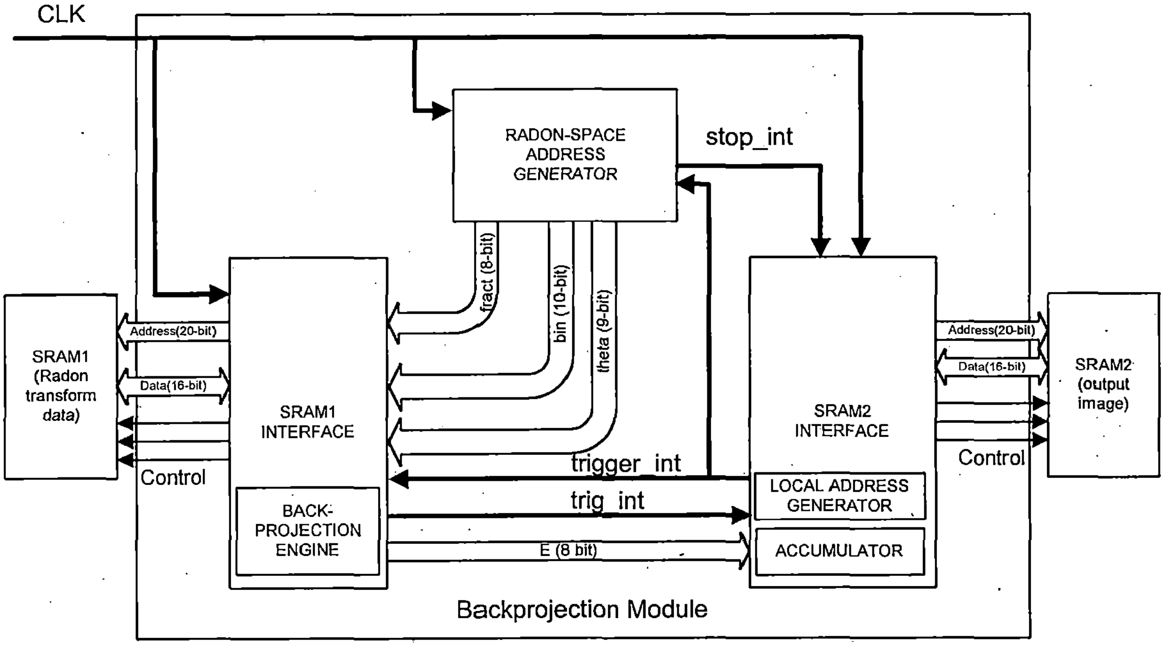


Figure 4.5: A Schematic diagram of backprojection hardware

The whole design is divided among three main modules, each of which in turn contains several sub-modules. Here SRAM1 is the external SRAM used to store the input data before processing starts. SRAM1 Interface circuit is a customized hardware designed to interface SRAM1 with hardware backprojection module. SRAM1 Interface circuit contains another sub-module backprojection engine as shown in Figure 4.5. This module generates the correct value of intensity, which has to be accumulated in the pixel being addressed currently by SRAM2 Interface. A simplified architecture of backprojection engine has been given in Figure 4.6. This sub-module takes two values from continuous memory locations in SRAM1 and generates an intermediate signal *E*.

This signal is passed to Accumulator sub-module of SRAM2 Interface. Accumulator adds the value of this signal to the value of *i(x,y)* and writes it back at the same location in SRAM2.

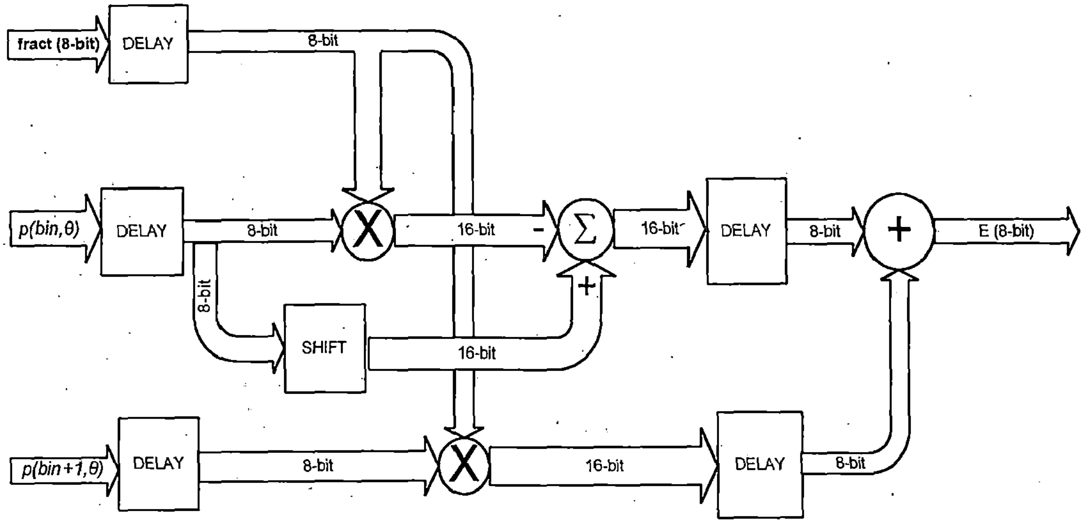


Figure 4.6: The architecture of Backprojection Engine

Both SRAM Interfaces interact with each other with the help of two handshaking signals *trigger_int* and *trig_int.* The Local Address Generator sub-module generates the addresses for SRAM2 in a raster-scan fashion. As soon as all projections are backprojected, the Radon-space address generation module sets the signal *stop_int* to high. This signal freezes the FSM running in SRAM2 Interface ensuring that there is no further communication with SRAM2. This in turn freezes the FSM running in SRAM1 Interface circuit because both are running in a handshaking mode. This in turn reduces the power loss in the circuit when it is in standby mode. For the complete implementation of Filtered backprojection algorithm, this backprojection circuit has to be embedded in a system, which provides these projections after filtering. A simple block diagram of such a system is shown in Figure 4.7.

The block diagram shown in figure 4.7 includes three additional modules. Here SRAM1 Interface directly reads the radon transform data one byte at a time and sends it to the input of FIR filter. The FIR filter has been designed using a direct implementation in VHDL. This filter is designed to accept one handshaking signal whenever a new data is provided at its data input. This filter has a latency of three clock cycles for a three-tap and a latency of four clock cycles for a seven-tap implementation. At the next positive going clock after latency period, this filter provides output for the current input. Along with the output, this filter also gives a positive pulse on the output signal RDY, indicating the presence of a valid output on the output port of the filter. This output is directly stored in an internal Block SRAM. The Block SRAM feature is provided by Xilinx Virtex series for addressing on-chip memory requirements problems generally faced in signal and image processing applications. This feature is available as an IP core, which can be embedded in a VHDL design using Xilinx Core-generator.

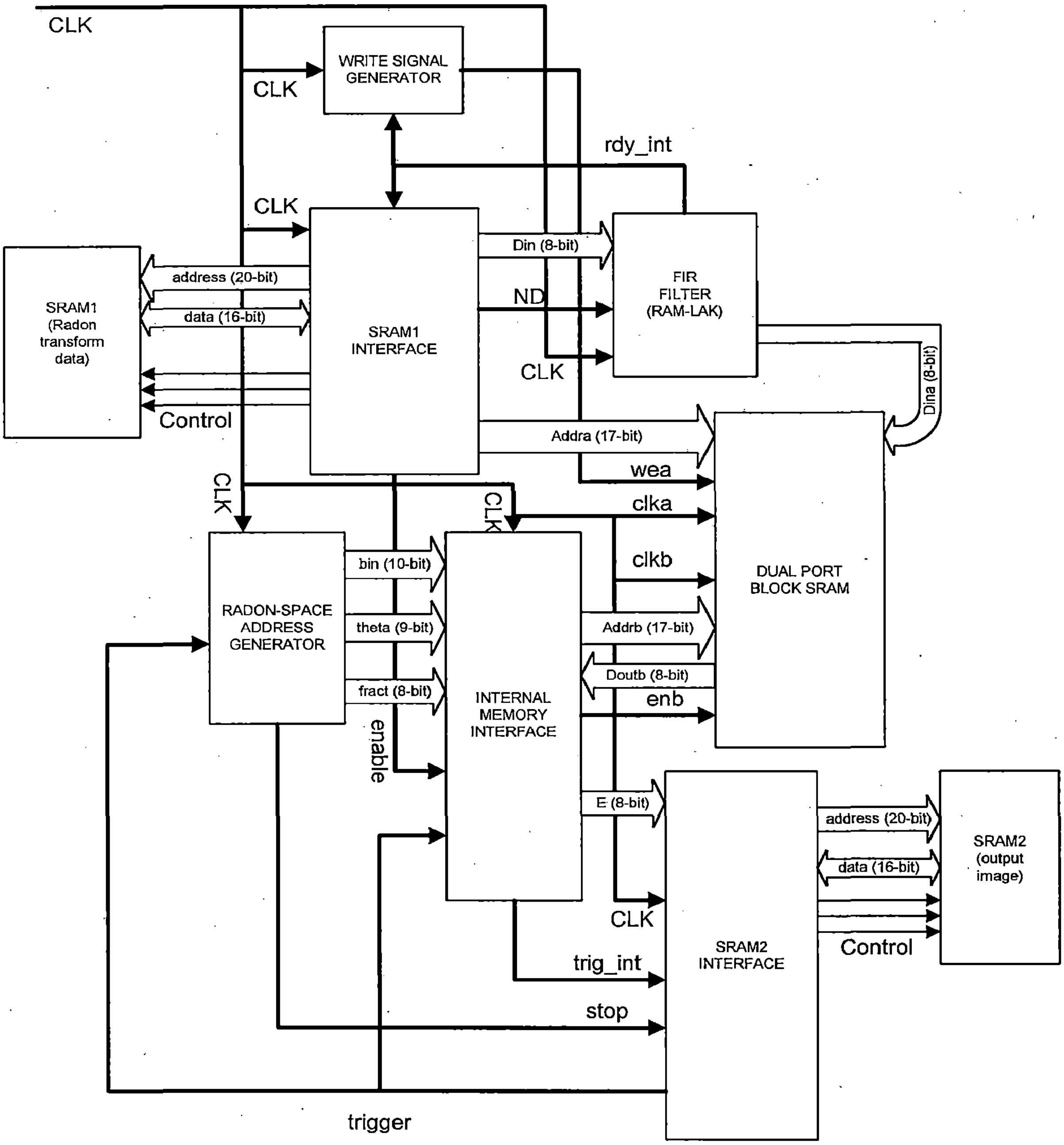


Figure 4.7: Block diagram for implementation of Filtered backprojection algorithm

In the current work, a dual-port Block SRAM core has been used. Dual-port SRAM provides the facility to access a single memory map simultaneously through two independent ports. Here one port is configured to work as write only and second one is configured to work as a read only port. As soon as the first projection has been filtered and the filtered result has been stored in the Block SRAM, SRAM1 Interface circuit generates an enable signal for internal memory interface circuit. This part of the design works exactly same as backprojection circuit shown in Figure 4.5. Since FIR filter is capable of generating output at a rate much faster than the rate of arrival of a new input, there is no risk of reading any corrupted data from Block SRAM. Therefore, the operations of filtering and backprojection may run in parallel safely, but definitely with a certain time lag. The proposed technique is a complete solution for image reconstruction problem since it only needs the projection data at its input and after processing output SRAM contains the accumulated image data.

4.6 CONCLUSION

Some general goals and requirements of a digital image processing systems are discussed briefly in this chapter. A stepwise algorithm for computation of Radon transform using a digital system has been formulated. A practical digital implementation of the Radon transform computation circuit is also proposed in support of the algorithm. The problem of Filtered backprojection has been discussed in detail from digital implementation point of view. An algorithm for the computation of backprojection from digital projection data is proposed here, which shows its direct correspondence with Radon transform algorithm discussed in section 4.2. Filtered backprojection algorithm has been implemented and discussed in detail. A novel approach has been proposed for digital implementation of FBP algorithm, where filtering and backprojection processes are made to run in parallel. This would reduce the computation time and hence the power requirements for the customized hardware.

CHAPTER - 5

DESIGN AND IMPLEMENTATION USING FPGA

5.1 INTRODUCTION

This chapter describes the design flow used to create simple and moderate level FPGA and ASIC devices. A design flow starts with a design specification. The designer then converts this design specification into a corresponding set of interconnected functional blocks, each block representing an elementary operation to be performed within the algorithm. This approach is sometimes referred to as Top to Bottom approach. The selection of Top to Bottom or Bottom to top approach depends highly on the designer's knowledge and interests.

After a design has been formulated, the design enters his design to a CAD (Computer Aided Design) system [13]. This step is known as design entry step. From this step onwards, the designer is there only for check and debugs purposes. A CAD system is a set of a number of CAD tools needed for a complete digital design. A CAD system typically includes tools for the following tasks: design entry, synthesis and optimization, simulation, and physical design.

VHDL is the language that has been used to model and design the electronics components used to build the designs in the current work. VHDL supports three types of modeling techniques namely given as:

- Structural modeling
- Dataflow modeling
- Behavioral modeling

In the current work all of these three modeling techniques are used. This type of modeling is often termed as mixed-style modeling. Behavioral modeling is used for designing the basic building blocks in the design. Then a structural modeling style is used to integrate those components and make a larger design module.

In this chapter, initially VHDL and related terminologies to VHDL language are introduced in brief. Next section gives a basic introduction about FPGA and related terminologies. The final section of the chapter is intended to give a special attention Xilinx Virtex II pro device and its features.

5.2 VHDL

VHDL is an acronym, which stands for VHSIC (Very High Speed Integrated Circuits) Hardware Description Language. Initially VHDL was developed for modeling purpose only. Today VHDL supports to a number of needs in a digital hardware design process. First, it allows description of the structure of a system, that is, how it is decomposed in subsystems and how those systems are interconnected. Secondly, it allows the specification of the function of a system using familiar programming language forms. Third, as a result, it allows the design of a system to be simulated before any hardware implementation. This enables a hardware designer to verify his code for a desired functionality at a very early stage. Fourth, it allows the detailed structure of a design to be synthesized from a more abstract specification, allowing designer to concentrate more on strategic design decisions.

VHDL is being used for documentation, verification and synthesis of large digital designs. This is one of the key features of VHDL, since the same code can be used for all three purposes, thus saving a lot of time and effort. Again VHDL supports three entirely different types of modeling styles which makes it able in modeling almost any digital operation easily. Following sections will introduce more features of the language by examining its use for each of these three design methodologies.

VHDL was established as the IEEE 1076 standard in 1987. In 1993, the IEEE 1076 standard was updates and an additional standard IEEE 1164 was adopted. In 1996, IEEE 1076.3 became the VHDL synthesis standard.

5.3 VHDL PROGRAMMING TECHNIQUES

Various methods can be used to write model for digital circuits using VHDL. Circuit to be designed can be modeled using its functional description known as behavioral programming, by describing the system with the help of its components known as structural programming or just by giving the flow of signals which is known as dataflow style of modeling. All these modeling styles can also be mixed to define the function of a design. This type is modeling is known as mixed style modeling.

5.3.1 Behavioral Programming

In VHDL, a description of the internal function of an entity is called the architecture of that entity. There may be a number of architectures for a single entity, corresponding to alternative implementations that perform the same function. an behavioral architecture can be written for the entity that describes the function of the entity in a abstract way. Such an architecture body contains only process statements, which are collection of actions to be executed in sequence. These actions are called sequential statements and are much like the kinds of statements we see in a conventional programming language. The type of actions that can be performed including evaluating expressions, assigning value to variables, conditional execution, repeated execution and subprogram calls. In addition, there is a sequential statement that is unique to hardware modeling languages, the signal assignment statement. This is similar to variable assignment, except that it does not assign the value immediately rather it schedules the assignment of that value to happen at a specific time.

5.3.2 Structural Programming

An alternative way of describing an entity is to specify its internal structure instead of its functionality. Internal structure of an entity is defined by an interconnection of the sub-systems whose behavior is known in advance. In structural modeling the architecture body contain the definition of the sub-systems as components and their interconnection. The architecture is divided into toe sections. One section is signal and component declaration section and another is the body of the architecture. The signal and component declaration section contain the definition of all components in the system and the signals which are used to interconnect those components. The body of architecture contains only component instantiation statements. These component instantiation statements connect the ports of components through some internal signals, rest ports may be directly connected to the ports of the entity itself. Also, more than one instances of any component defined in the comment declaration section may be used inside the body of the architecture.

5.3.3 Dataflow Modeling

Dataflow modeling also do not define the functionality the circuit directly. This type of modeling uses a set of concurrent signal assignment statements. However, the functionality of the circuit designed using this style of modeling can be understood implicitly by looking at the flow of data. Therefore, the name dataflow modeling is used for this style of modeling.

5.3.4 Mixed Mode Modeling

A model of a circuit is not restricted to be defined by using only one type of modeling style. Generally, circuits are defined by an interconnection of few sub-systems and few processes running within the current design. In this case, signals are used for connecting component instances as well as inside the processes for sequential signal assignments.

In a mixed mode, modeling the component instances and the processes runs independent to each other. In other words, it is said that these processes and components run concurrently.

5.4 TERMINOLOGIES IN VHDL

VHDL is a worldwide standard for the description and modeling of digital hardware. VHDL provides a set of options for describing hardware. In addition to the simpler language construct, there are many user friendly and powerful tools available in the market to support VHDL based hardware modeling and synthesis. Every VHDL tool available in the market atleast supports the packages and libraries defined in the IEEE standard. These packages and libraries contain various fundamental components and functions, definition of different data types supported by standard VHDL similarly as any standard high-level language compiler.

VHDL has ample amount of features appropriate for describing the behavior of electronic components ranging from simple logic gates to complete microprocessor, high-speed digital signal processor and custom chips. Features of VHDL allow controlling the timing attributes of the signal and using them for conditional execution of statements.

5.4.1 Entity

This is the basic unit of a VHDL description which gives the input and output ports of the digital circuit being modeled. It also defines the types of these ports. For example, if entity with name XYZ has input ports A, B of type bit and a single output port C of type bit is to be modeled then, in VHDL this entity will be defined as:

```
Entity XYZ is
    Port (A: in bit;
          B: in bit;
          C: out bit);
```

End XYZ;

5.4.2 Packages

Packages are intended to hold commonly used declarations such as constants, type declarations and global subprograms. Packages can be included within the same source file as other design units or may be placed in a separate source file and compiled into a named library. The later method is more useful while designing large systems. A package compiled in a separate file may be used in different modules of the same project. The IEEE 1164 standard provides a standard package named std_logic_1164 that includes the declarations for the types: std_logic, std_ulogic, std_logic_vector and std_ulogic_vector. This package also contains many functions for these data types.

5.4.3 Design Libraries

A design library is an implementation-dependent storage facility for previously analyzed design unit. This results in many different implementations in synthesis and simulation tools. In general, however, design libraries are used to collect commonly. In general, however design libraries are used to collect commonly used design units into uniquely named areas that can be referenced from multiple source files in a design.

5.4.4 Components

Components are used to connect multiple VHDL design units together to form a larger, hierarchical design. Using hierarchy can dramatically simplify the design description and can make it much easier to re-use portions of the design in other projects. Components are also useful while making the use of third party design units, such as simulation models for the standard parts, or synthesizable core models obtained from an organization specializing in such models.

5.4.5 Configurations

Configurations are features of VHDL that allow large, complex design descriptions to be managed during simulation. This feature of VHDL is not supported for synthesis. One example of how to use a configuration is to construct two versions of a system level design, one of which makes use of high-level behavioral description of the system components, while a second version substitutes in a post-synthesis timing model of one or more components.

For large projects involving many engineers and many design revisions, configuration can be used to manage versions and specify how a design is to be configured for system simulation, detailed timing simulation and synthesis. Because simulation tools allow configuration to be modified and recompiled without the need to recompile other design units, it is easy to construct alternative configurations of a design very quickly without recompiling the whole design.

5.5 FPGA

A field programmable gate array is a semiconductor device containing programmable logic components called "logic blocks" and programmable interconnects. Programmable logic blocks can be programmed to perform the function of basic logic gates such as AND, and OR, or more complex combinational functions such as multiplexers or simple mathematical functions. In most of FPGAs also contain memory elements, which may be simple flip-flop or more complex blocks of memories.

A hierarchy of programmable interconnects allows the logic blocks to be interconnected as needed by the system designer. Logic blocks and programmable interconnects can be programmed by customer/designer after FPGA has been manufactured, to implement any logic function. Because of the feature of programmability in field it is called field programmable gate array.

FPGAs are usually slower than their application-specific integrated circuit (ASIC) counterparts, because they are designed to be flexible at the cost of performance. Also FPGAs draw relatively more power than their ASIC counterparts. However, FPGAs have some very strong benefits against ASICs. FPGAs take a much shorter time to market, they can be reprogrammed in field to fix the bugs and FPGA based designs have a low non recurring engineering (NRE) cost.

FPGAs were first introduced in the mid 1980s to replace multi-chip glue logic circuits with a single reconfigurable solution. FPGAs have far overgrown their sole use as a replacement for simple glue logic circuits. Presently, FPGA applications include signal and image processing, graphic accelerators, military target correlation/recognition, cryptography, reconfigurable computation, on-chip emulation, etc...

Internal structures of FPGA are commonly constructed using a symmetric tile structure containing a network of switchboxes, logic boxes, wire channels and input-output blocks. A switchbox (SB) is a location in the FPGA fabric that provides a method to connect the internal wires together. The switchbox allows horizontal wire segments to switch to the vertical wires. The size and contents within a logic block vary greatly depending on the manufacturer and the target market. For example, FPGAs targeted towards cost-effective solution typically contain simpler logic block than an FPGA targeted for high performance applications. Although contents within a logic block may vary for different architectures, there are two basic building blocks found in a logic block: memory elements and function generators. Memory elements provide designer the facility to store information temporarily until desired conditions are met. Function generators can be programmed to generate any function up to the number of inputs into the function generator. Depending on the architecture, some function generators can operate in different modes such as random access memory (RAM), read only memory (ROM), or even more modes like shift registers. FPGAs are configured through a bitstream that is loaded into the device in order to program it. A bitstream is a file created by the CAD tool which configures the switchboxes, logic blocks and other FPGA resources.

FPGAs have redefined the boundaries if the digital electronics allow designers to build systems piecewise. Multiple designers can rapidly test and verify the functionality of each individual piece of a system to ensure proper functionality prior to merging the entire system together. With increasing interest in reconfigurable computing, FPGAs are now recognized as most viable and cost-effective solution. Whether a design is statically or dynamically reconfigurable, a FPGA provides rapid programmability and short time to market design cycle. In the current work a platform FPGA provided by Xilinx is being used for final implementation.

5.6 XILINX VIRTEX II PRO PLATFORM FPGA

Xilinx is the major player in the market of reprogrammable devices. It is known for providing devices with most advanced features. Xilinx Virtex-II Pro and Virtex-II Pro X devices fall in the category of platform FPGA. These devices are dedicated for those designs that require IP core and customized modules. This family incorporates multi-gigabit transceivers and PowerPC CPU blocks in the Virtex-II Pro series FPGA architecture. These FPGAs are targeted for high-end digital applications such as telecommunication, Digital Signal and Image processing, video processing, wireless, and networking.

This device family is manufactures with a leading edge 13-micrometer CMOS technology with a nine layer copper mask printed circuit board (PCB) technology. Virtex-

II Pro architectures are optimized for high performance applications. Combining a wide variety of flexible features and IP cores, Virtex-II family enhances programmable logic design capabilities in a powerful alternative to mask programmed gate arrays.

Virtex-II Pro devices are user programmable gate arrays with various configurable elements and embedded blocks optimized for high density and high-performance system designs. Virtex-II Pro devices provide following functionalities

- Embedded high-speed serial transceivers enable data bit rate up to 3.125 Gb/s per channel (RocketIO).
- Embedded IBM Power PC 405 RISC processor blocks provide performance up to 400 MHz.
- SelectIO-Ultra blocks provide the interface between package pins and the internal configurable logic. Most popular and leading edge IO standards are supported by programmable IOBs.
- Configurable Logic Blocks (CLBs) provide functional elements for combinatorial and synchronous logic, including basic storage elements. BUFTs (3-state buffers) associated with each CLB drive dedicated segmentable horizontal routing resources.
- Block SelectRAM+ memory modules provide large 18 kb storage elements of true Dual-Port RAM.
- Embedded multipliers are 18bitx18bit dedicated multipliers
- Digital Clock Manager (DCM) provide self calibrating, fully digital solution for clock distribution delay compensation, clock multiplication and division, and coarse- and fine-grained clock phase shifting

A new generation of programmable routing resources called Active Interconnect Technology (AIT) interconnects all these elements. The general routing matrix (GRM) is an array of routing switches. Each programmable element is tied with a switch matrix, allowing multiple connections to the GRM. The overall programmable interconnection is hierarchical and supports high-speed designs.

All programmable elements, including the routing resources, are controlled by values stored in static memory cells. these values are stored in memory sells during configuration and can be reloaded to change the functionality of the logic elements.

In the current work, two of these advanced features are used. A Block SelectRAM+ is used to store the result from filtering stage. In addition, an IP core was used to generate trigonometric sine and cosine functions. These IP cores are embedded in the VHDL design with the help of Xilinx CORE Generator System.

5.7 Xilinx CORE Generator System

The CORE Generator System is a design tool that delivers parameterized cores optimized for Xilinx FPGAs. It provides you with a catalog of ready-made functions ranging in complexity from simple arithmetic operators such as adders, accumulators, and multipliers, to system-level building blocks such as filters, transforms, FIFO, and memories. The CORE Generator System creates customized cores that deliver high level of performance and area efficiency. This is accomplished by taking advantage of Xilinx core friendly FPGA architecture and Xilinx Smart-IP technology.

Xilinx Smart-IP technology provides the architectural advantages such as look up tables, distributed RAM, block RAM, embedded multipliers and segmented routing. This technology also enables relative location constraints, and expert logic mapping and floorplanning to optimize performance of a given core instance in a given Xilinx FPGA architecture. Smart-IP technology provide following beneficial features:

- Optimization of physical layout for high performance
- Prediction for the resource utilization and performance
- Optimize the power consumption by obtaining a compact design and interconnect minimization
- Performance independent of target device size
- Ability to use multiple instances of an IP, in the same design, without any deterioration in performance
- Provides flexibility in design
- Ability to make trade-offs between performance and size of the design

The designs in the current work are using following two IP cores:

- Sine/Cosine Look-Up Table
- Dual-Port block memory

The Sine/Cosine module accepts an unsigned input value THETA and produces two's complement output of SINE(THETA) and/or COSINE(THETA). This IP care is completely parameterized and one can easily control the input THETA width as well as output SINE and/or COSINE width values. The relation between integer input THETA and actual radian angle θ is given by:

$$\theta = THETA * \frac{2\pi}{2^{THETA\_WIDTH}}\ radians \tag{5.1}$$

The core computes $sin(\theta)$ and $cos(\theta)$ and presents the two's complement on the output ports SINE and COSINE respectively. The outputs are presented as fractional fixed-point data.

The Dual-Port Block Memory module for Spartan-II and Virtex is composed of single or multiple 4kb blocks caked SelectRAM+. The Dual-Port Block Memory module for Virtex-II, Virtex-II Pro, Virtex-4 and Spartan-3, is composed of single or multiple Virtex-II 18kb blocks enabling deeper and wider memory implementations. These memory cores offer fast, discrete, and large blocks of memory in the devices from Virtex family. A Dual-Port memory module has two independent ports that enable shared access to a single memory map and which are generated on the basis of parameters selected by user. Both ports are functionally identical, with each port providing read and write access to the memory.

5.8 CONCLUSION

Programming language used to design the hardware and related terminologies has been discussed in detail. A justification for using VHDL is given by emphasizing the advantages of using this Hardware Description Language (HDL). The type of hardware used in the current work has been introduced and the significance of using FPGA has been explained. In the current work, a video board developed around a Xilinx Virtex-II Pro FPGA chip is being used. Therefore, an overview of Xilinx Virtex-II Pro family FPGA has been given along with the key features provided to the user. Finally, a Xilinx CORE generator system has been discussed that has been used in current work to embed two Xilinx IP cores in the current design.

CHAPTER - 6

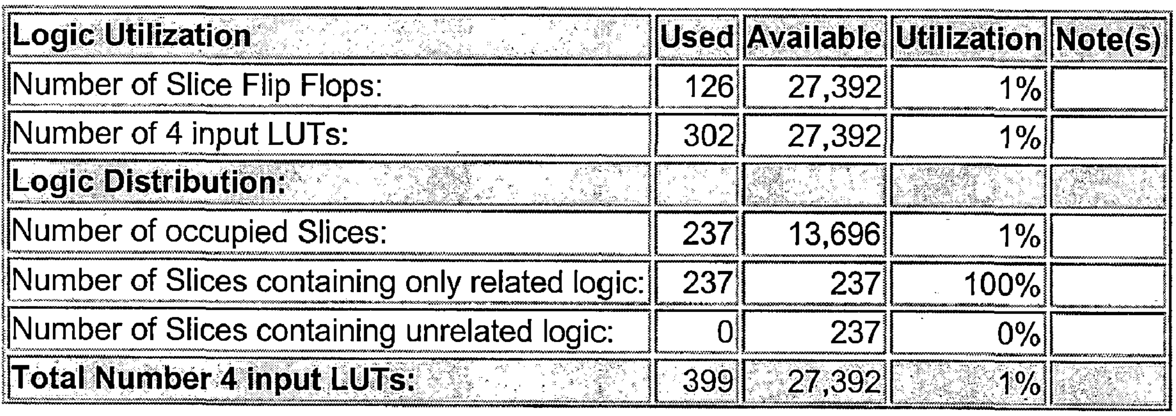
RESULTS AND CONCLUSION

6.1 RESULTS

All the basic entities in the design of Radon transform Circuit as well as Filtered backprojection circuit are programmed using behavioral model. In addition, the Radon transform circuit also uses one Xilinx IP core for generating the values of trigonometric terms. The Filtered backprojection circuit uses two Xilinx IP cores, one for trigonometric terms and second is a dual-port SelectRAM for storing the results of filtering. FIR filter has been designed using fixed coefficients. The impulse response of FIR filter has been implemented using the Direct VHDL implementation. The symmetry and pipelining features are extracted to improve the performance of the filter.

Table 6.1 and 6.2 are given here to show the resource utilization for the both designs. Filtered backprojection hardware utilizes a greater amount of resources because it needs to implement three extra modules for filtering the projection and storing it. From Table 6.2, one more information is noticeable that, for storing the filtered projections for a 128x128 pixel input image about 52% of the on-chip memory resources are consumed. Therefore, with the increasing size of the input image a larger amount of on-chip memory will be required. One more noticeable point is the number of 4-input Look-Up tables (LUTs) required to design the other logic elements. For calculation of Radon transform less that 2% of the available LUTs are requited, whereas for Filtered backprojection circuit it is more than 3%.

Table 6.1: Device Utilization Summary for Radon transform Circuit



Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Flip Flops:	126	27,392	1%	
Number of 4 input LUTs:	302	27,392	1%	
Logic Distribution:				
Number of occupied Slices:	237	13,696	1%	
Number of Slices containing only related logic:	237	237	100%	
Number of Slices containing unrelated logic:	0	237	0%	
Total Number 4 input LUTs:	399	27,392	1%	

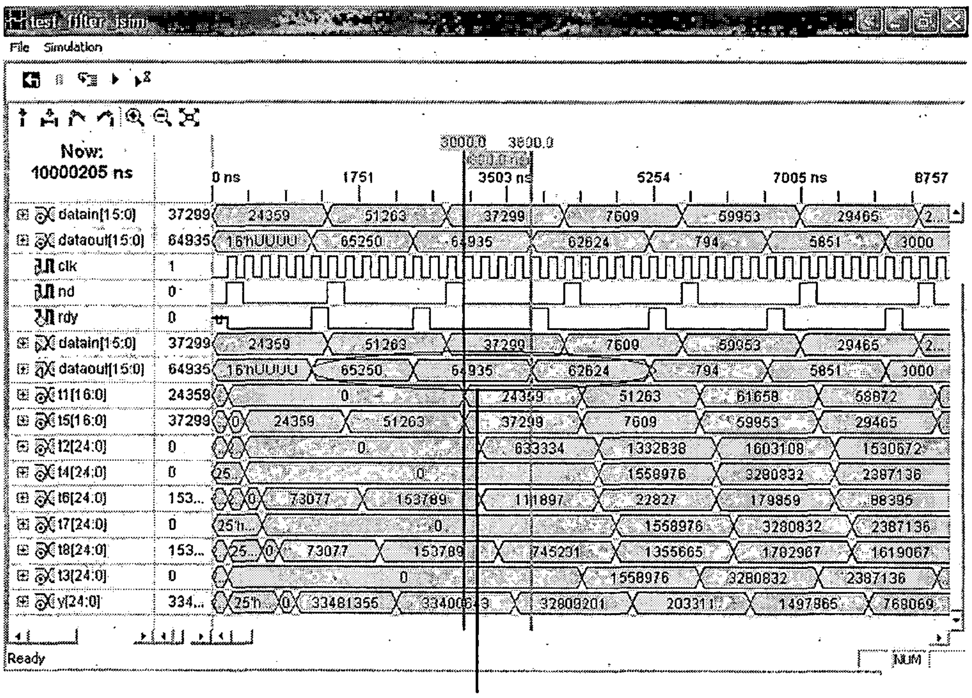
Number used as logic:	302			
Number used as a route-thru:	97			
Number of bonded IOBs:	87	644	13%	
Number of PPC405s:	0	2	0%	
Number of Block RAMs:	1	136	1%	
Number of MULT18X18s:	8	136	5%	
Number of GCLKs:	1	16	6%	
Number of GTs:	0	8	0%	
Number of GT10s:	0	0	0%	
Number of RPM macros:	1			

Table 6.2: Device Utilization Summary for Filtered backprojection circuit

Logic Utilization	Used	Available	Utilization	Note(s)
Total Number Slice Registers:	196	27,392	1%	
Number used as Flip Flops:	195			
Number used as Latches:	1			
Number of 4 input LUTs:	709	27,392	2%	
Logic Distribution:				
Number of occupied Slices:	466	13,696	3%	
Number of Slices containing only related logic:	466	466	100%	
Number of Slices containing unrelated logic:	0	466	0%	
Total Number 4 input LUTs:	835	27,392	3%	
Number used as logic:	709			
Number used as a route-thru:	126			
Number of bonded IOBs:	72	644	11%	
Number of PPC405s:	0	2	0%	
Number of Block RAMs:	72	136	52%	
Number of MULT18X18s:	11	136	8%	
Number of GCLKs:	1	16	6%	
Number of GTs:	0	8	0%	
Number of GT10s:	0	0	0%	

Figure 6.1 shows the functional simulation of the filter block of the Filtered backprojection hardware, for a seven-tap impulse response implementation. For an odd number of taps *n*, an FIR filter produces *(n-1)/2* transients outputs at the start of output

stream and same number of transients outputs at the end of the stream. It is shown in Figure 6.1, the output of the FIR filter contains three transient output at the start of its output sequence. There are two handshaking signals *nd* and *rdy* to control the operation of the filter. From the Figure 6.1, it is clear that *rdy* goes high exactly after four clock cycles from the instant when signal *nd* was high. A positive pulse on input signal *nd* indicates a new data input to the filter and *rdy* signifies that corresponding output is ready to be readout at the filter output.

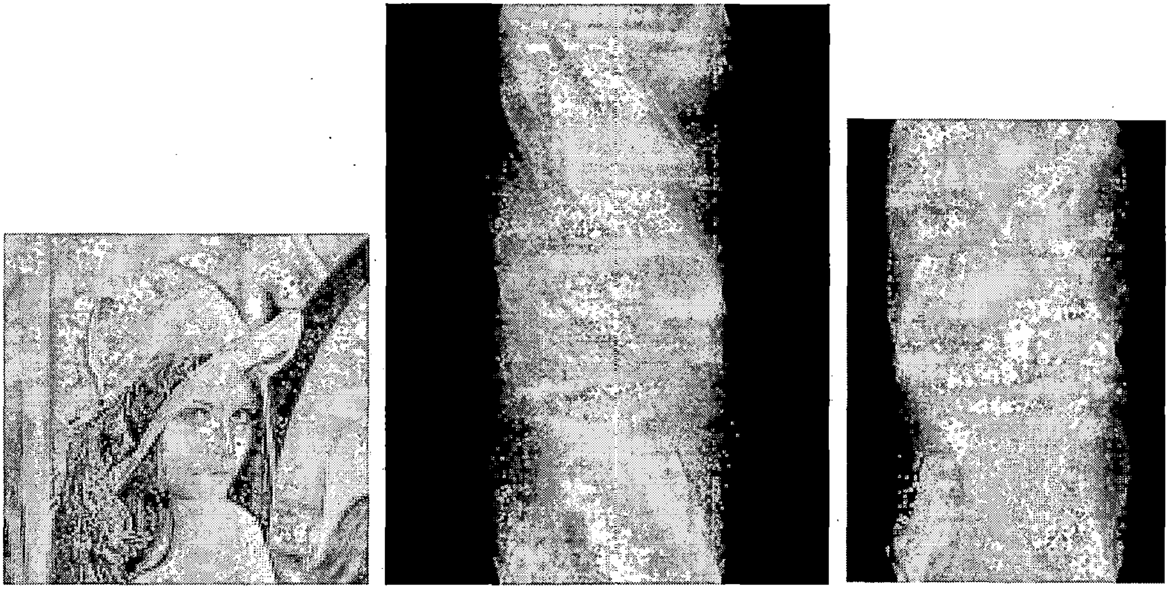


First three transient outputs for a seven-tap implementation

Figure 6.1: Simulation results for a seven-tap Ram-Lak FIR filter

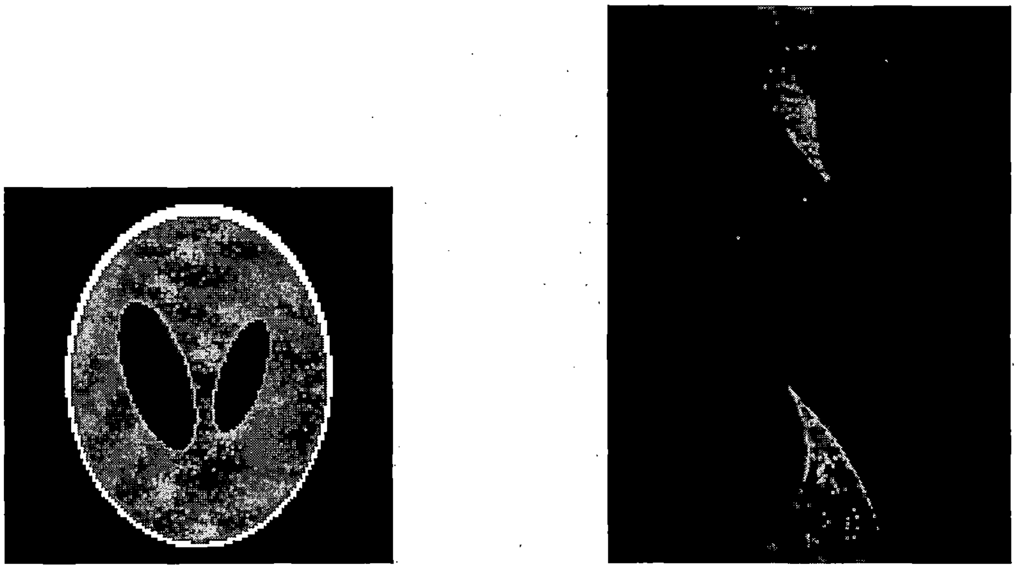
Results from Radon transform Circuit:

This section shows the experimental results obtained for the Radon transform computation circuit. Though there is no standard format given for the presentation of the radon transform, a convention similar to MATLAB has been adopted in current work, where origin of the image is assumed to be located at the center of the image space. In Figure 6.2, the radon transform for the Lena image is given using designed hardware and a MATLAB result for the same input has been presented for visual comparison.



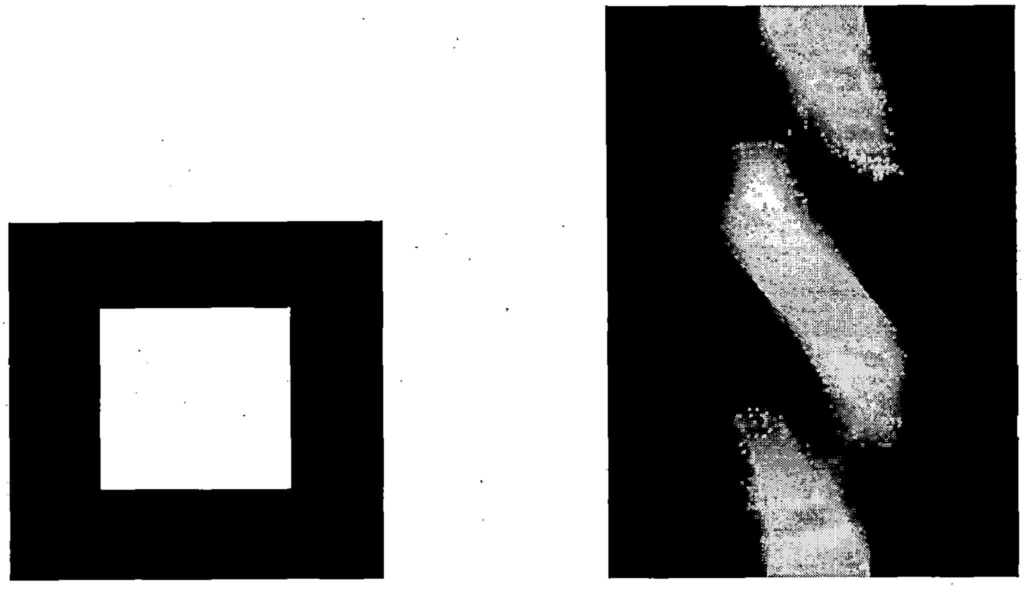
(a) (b) (c)

Figure 6.2: (a) A standard test image of Lena (128x128 Bitmap), (b) The Radon transform of (a) computed by designed hardware, (c) The Radon transform of (a) obtained from MATLAB



(a) (b)

Figure 6.3: (a) An approximate Shepp-Logan phantom (128x128 Bitmap), (b) Radon transform of (a) computed by designed hardware

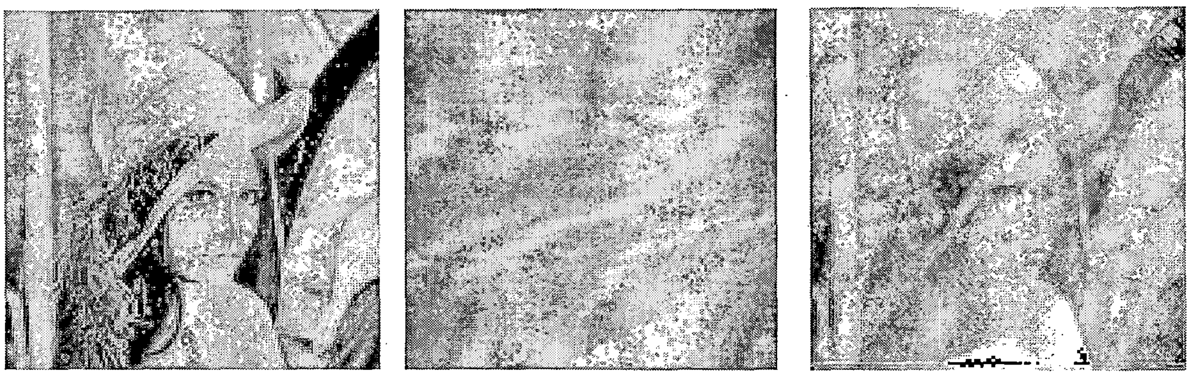


(a) (b)

Figure 6.4: (a) An example of very high contrast image (128x128 Bitmap), (b) Radon transform of the image given in (a), computed by designed hardware

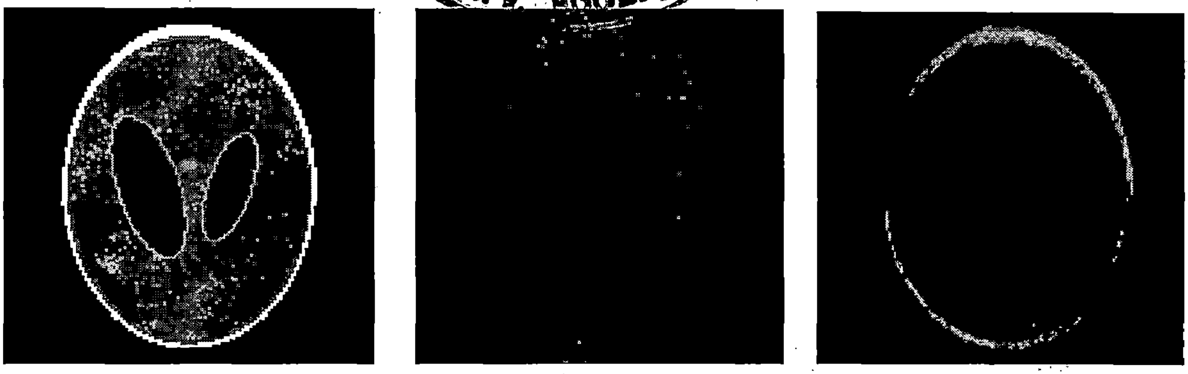
Result from Backprojection Circuit:

The reconstructed images for different types of input images are given here, which has been computed with the help of the current hardware. Three different images taken in the case represent different types of object densities. The results for backprojection without filtering and Filtered backprojection computations are given in Figures 6.5, 6.6, and 6.7.



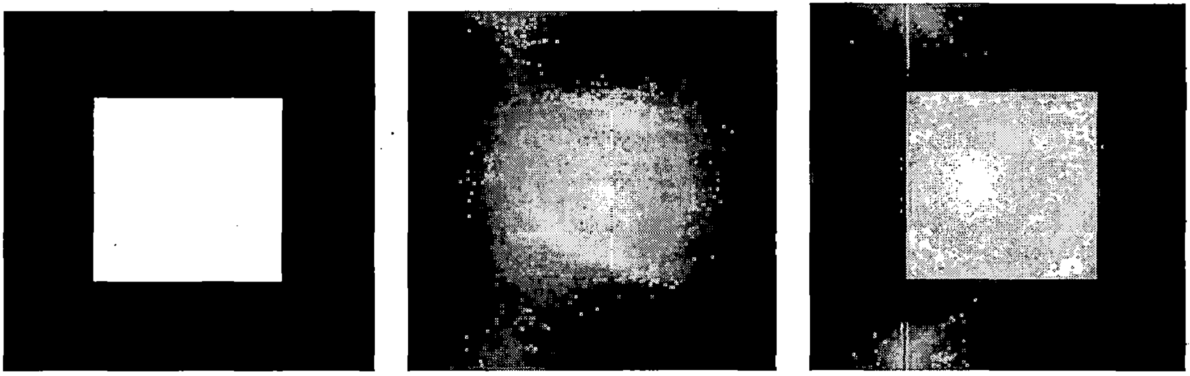
(a) (b) (c)

Figure 6.5: (a) input Lena image, (b) Blurred reconstructed image of (a) computed by backprojection module (without filtering), (c) Reconstructed image using Filtered backprojection hardware



(a) (b) (c)

Figure 6.6: (a) Input Shepp-Logan image, (b) Blurred reconstruction of (a) computed by backprojection module (without filtering), (c) Reconstructed image using Filtered backprojection hardware



(a) (b) (c)

Figure 6.7: (a) Input high-contrast image, (b) Blurred reconstruction of (a) computed by backprojection module (without filtering), (c) Reconstructed image from Filtered backprojection hardware

6.2 CONCLUSION

From the results obtained for different images, from Radon transform and Filtered backprojection circuit, are self-explanatory for the successful implementation of both the circuits. The computation of Radon transform requires a very nominal amount of available resources for implementation. Generally, multiplier circuits need a large amount of logic resources for their implementation. Therefore, in the current work the required hardware is reduced by utilizing the on-chip dedicated multipliers provided on Xilinx Virtex-II Pro FPGA chip. The Xilinx IP cores provided by platform FPGA architecture allow an easy calculation of the complex trigonometric functions on-chip, which in turn is useful for on-chip address and interpolation factor generation. 'Place and Route static timing report' suggests that maximum operating frequency for the Radon transform computation circuit is around 54 MHz. However, for the filtered backprojection computation circuit, it is around 37 MHz. The operating speed of filtered backprojection computation circuit is slower than that of Radon transform computation circuit. In the proposed approach for filtered backprojection, the filtering and backprojection steps are performed as single step. Since current approach uses on-chip dual-port memory, the latency of data transfer operation from filtering stage to backprojection stage has been reduced significantly. Therefore, this operating frequency accounts for the time delay for filtering as well as backprojection operation. Since, this proposed approach merges two separate steps into a one single step; this is a faster implementation over conventional reconstruction circuits.

CHAPTER - 7

FUTURE SCOPE

In the current work, emphasis is placed on designing the classical approach of Filtered backprojection algorithm. However, based on current work a much deeper attention may be paid to achieve parallelism in the basic algorithm. Other aspects like device utilization on FPGA, power consumption can be taken in consideration and methods can be explored to minimize them. Few suggestions under which the current work may be improved are given as:

- Since algorithm involves complex calculations, software-hardware decomposition of the algorithm may give a better possibility. On-chip PowerPCs can be utilized for running the software code.
- Current design can be extended to develop reconstruction hardware for more recent generation CT scanners.
- Extension of the present work in three dimensional image reconstruction problems can be done by adding modules for stacking two-dimensional sectional images.

[14] A. Kingston and Imants Svalbe, "Mapping between Digital and Continuous Projections via the Discrete Radon transform in Fourier Space," Proceedings of 7th Digital Image Computing: Techniques and Applications, 10-12 Dec, 2003, Sydney

[15] D. Soudris, T. Chronopoulos, N. Koussaoulas, and T. Stouraitis, "Systematic Design of Novel Architectures for implementation of Radon Transform," Proceedings of the 38th Midwest Symposium on Circuits and Systems, 1995.

[16] Srdjan Coric, Miriam Leeser, Erric Miller, "Parallel-Beam backprojection: An FPGA Implementation optimized for Medical Imagin," Journal of VLSI Signal Processing Systems, Volume 39 , Issue 3 (March 2005), Pages: 295 - 311

[17] Srinivasa, N and Krishnan, V and Ramakrishnan, KR and Rajgopal, K, "Image Reconstruction Form Truncated Projections: A Linear Predication Approach," Proceedings 1986 IEEE International Conference on Acoustics, Speech, and Signal Processing. ICASSP '86, pages Vol.11, 1733-1736, Tokyo.

[18] G. N. Ramachandran and A. V. Lakshminarayanan, "Three-Dimensional Reconstruction from Radiographs and Electron micrographs: Application of convolution instead of Fourier transforms," Proceedings of National Academy of Sciences U. S., 68:2236-2240,1970.

[19] Harish P. Hiriyannaiah, "X-ray Computed Tomography for Medical Imaging," IEEE signal processing magazine, 1053-5888, 1997.

[20] Xilinx Support: Links to documentation by device support, Support Quicklinks, and Support News. [online]Available: www.**xilinx**.com/support/mysupport.htm

[21] FPGA Co-processing Solutions for High-Performance Signal Processing Applications, [Online] Available: http://www.altera.com/

OTHER USEFUL REFERENCES:

[22] Stephen Brown, Zvonko Vranesic, "Fundamentals of Digital Logic with VHDL Design," Tata McGraw-Hill Edition 2002.

[23] Douglas L. Perry, "VHDL: Programming by Experience," Tata McGraw-Hill, Fourth Edition 2002.

[24] Wayne Wolf, "FPGA Based System Design," Pearson Education, Second Edition 2003.

[25] http://en.wikipedia.org

[26] www.ndt-ed.org/EducationResources

APPENDIX A

Virtex-II Pro – PCI Video Board

A.1 FPGA

Xilinx Virtex- II pro - XC2VP30 / XC2VP40/XC2VP50 device in FF1152 package. These are platform FPGAs that are based on IP cores and customized modules, optimized for high density and high performance system design. They empower complete solutions for telecommunication, wireless, networking, Video and DSP applications.

The family incorporates multi gigabit transceivers using RocketIO technology and 32-bit microprocessors (IBM's power pc 405 CPU) in the FPGA fabric. The PowerPC core is a 0.13 µm implementation of the IBM PowerPC 405D4 core, with the ability to operate at 300+ MHz while maintaining low power consumption. Specially designed interface logic integrates the core with the surrounding CLBs, block RAMs, and general routing resources.

A.2 PowerPC 405 Processor

The PPC405 RISC CPU can execute instructions at a sustained rate of one instruction per cycle. On-chip instruction and data cache reduce design complexity and improve system throughput.

Within the Processor Block, there are four components:

1) Embedded IBM PowerPC 405-D5 RISC CPU core

Embedded PowerPC 405 (PPC405) core operating at 300+ MHz maintains low power consumption.

Specially designed interface logic integrates the core with the surrounding CLBs; block RAMs, and general routing resources.

Embedded PowerPC 405 core consists of the following functional blocks

- Instruction & Data Cache units
- Memory Management unit
- Fetch & Decode unit
- Execution unit
- Timers
- Debug logic unit

2) On-Chip Memory (OCM) controllers and interfaces

The OCM controllers (DSOCM & ISOCM) serve as dedicated interfaces between the block RAMs in the FPGA fabric and OCM signals available on the embedded PPC405 core.

ISOCM controller provides an interface to the 64-bit Instruction-Side Block RAM (ISBRAM) while

DSOCM controller provides an interface to the 32-bit Data-Side Block RAM (DSBRAM).

A.3 Clocks

User has option of using up to 2-different clock sources on the board which provide all necessary clocks for User logic and PowerPC

The clock sources provided on board are as follows

1. **Clock Source1**

32 MHz clock oscillator – supplied as standard and can be used as system clock for PPC.

2. **Clock Source2**

Socket for user clock source (foot print compatible with clock sources from 40 to 300 MHz)

User can assemble a clock oscillator of his choice if his system frequency is different from 40 MHz.

It should be noted that all the clock sources are connected to the global clock inputs. Thus user can use these external clocks as input to the on-chip DCM's.

A.4 Flash PROM

1.5 MB of Flash PROM is provided as standard, using three 512K X 16 PROMs. Can be upgraded to 3 MB, as these proms are footprint compatible with 1M X 16 PROMs.

Note - This upgradation is possible only during manufacturing

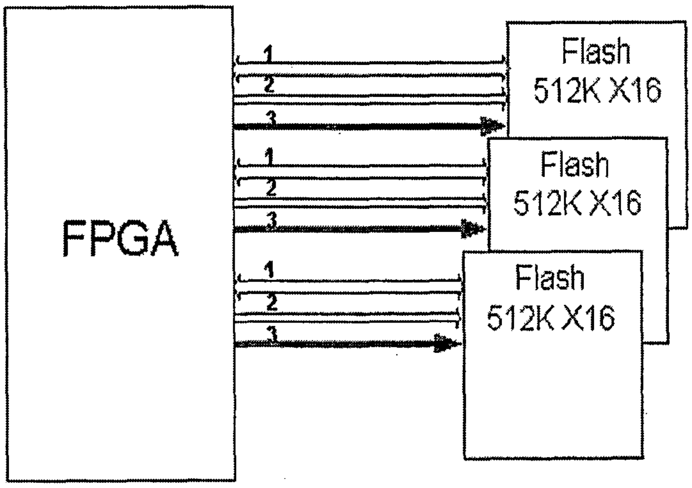


Figure A.1: Flash memory on Board

A.5 SRAM

Five 1M X 16 SRAM devices are independently interfaced to the on board FPGA.

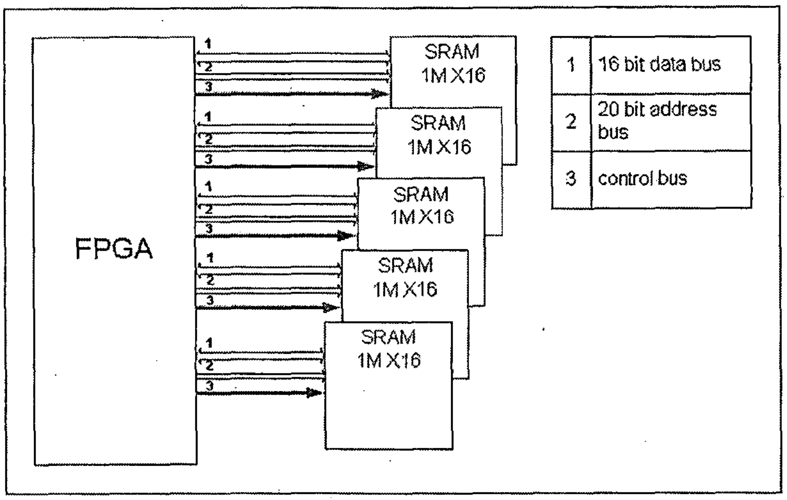


Figure A.2: SRAM on Board

A.6 RS232 Port

RS232C compatible connectivity is provided using device MAX3223. Signals provided are Rx, Tx RTS and CTS. These signals are terminated on a 10-pin FRC connector on board and a 10 pin FRC to 9-pin D connector interface cable is provided as standard accessory with the board. It is compatible with the UART core provided by the Xilinx EDK.

A.7 MIL- STD 1553

Mil standard 1553 defines a serial asynchronous data bus on which the messages are multiplexed among users. The transmission medium is a twisted wire cable. The standard specifies all electrical characteristics of transmitter, receiver and cable, and complete protocol. It requires a centralized control bus. HI-1573 transceiver device and PMDB27455 isolation transformer designed for MIL – STD 1553 by Holt integrated circuits are provided on board. Transmitter and receiver are provided with two pin berg header on the video board.

A.8 PCI Interface

A 32 bit, 33 MHz interface PCI interface is provided using PCI 9054 bus master device, with DMA capability. Drivers for Windows operating system are provided with the board.

A.9 Analog Input

One analog input channel is available with the following specifications

- ADC - AD9240.
- Resolution – 14 bits
- Max Sampling rate – 10 MSPS
- Input range - 0 to 5 Volts, single ended
- Input buffer – using AD8052 op-amp.
- Connector type – SMA

AD9240 is useful in various applications such as imaging, communications, and medical and data acquisition systems.

A.9.1 Video Input

One video input channel is available with the following specifications

- Video ADC. - TLV5734
- Video signal format – NTSC / PAL compliant RGB /YUV component video signal.
- Resolution – 8 bit
- Sampling rate –30 MSPS maximum.
- Input range - 1 Vp-p.
- Input buffer – using AD8052 op-amp.
- Connector type – SMA
- Selectable clamping function for RGB/YUV applications.

- Selectable Output Data Format for 4:4:4 (RGB, YUV), 4:2:2 and 4:1:1 (YUV) Format

A.9.2 Video Output

Two video output channels are provided on board with the following specifications

- Video DAC - THS8133
- Video signal format – NTSC / PAL compliant RGB /YUV component video signal.
- Resolution – 10 bit
- Sampling rate –80 MSPS maximum.
- Connector type – SMA

THS8133 provides current output; these current outputs can be converted into NTSC/PAL standard voltage levels by connecting a double terminated 75 ohms load.

- **Sync Generation** -Using sync, sync_t control signals, video sync signals can be added either on AGY (G/Y) channel or on all three channels with 7:3 video/sync ratio. Depending on the timing control of these signals both horizontal and vertical sync signals can be generated as well as either bi-level negative going or tri level pulses can be generated.
- **Blanking Generation** - An additional control input BLANK is provided that will fix the output amplitude on all channels to the blanking level. The absolute amplitude of the blanking level with respect to active video is determined by the GBR or YPbPr operation mode of the device.

A.9.3 Histogram Equalizer

The on board histogrammer LF48410 is capable of generating histograms and cumulative distribution functions of video images. It provides following features-

- 40 MHz data input and computation rate.
- 1024 X 24 bit memory array
- Histograms of images up to 4k X 4k with 10 bit pixel resolution.
- User programmable modes – Histogram, histogram accumulate mode, look-up table mode, Delay memory, single port memory.

A.9.4 Sync Separator-

Sync separator EL4583 used on board extracts timing from video sync in NTSC, PAL, and SECAM systems.

- Sync Separator – EL4583
- Input voltage range – 0.5 V to 2 V_{p-p}.
- Output signals –composite sync signal, vertical sync signal, horizontal sync signal, burst signal, odd/even signal, no signal detect output.
- Connector type – SMA

A.9.5 Digital IOs

Maximum of 16 true bidirectional IOs are available when using XC2VP30 device. (32 with XC2VP40 and XC2VP50).

These IOs are provided on two FRC connectors as follows,

1. **I/O Connector1**

A 20-pin box type FRC connector– provides 16 single ended, **5V tolerant** IOs.

These IOs support LVTTL, LVCMOSI_33 signaling standard **(3.3 V tolerant).**

2. **I/O Connector2**

A 20 pin box type FRC connector– provides 16 single ended, **5V tolerant** IOs.

These IOs support LVTTL, LVCMOSI_33 signaling standard. 5V tolerance is achieved by using quick switches for isolating the User I/O's and FPGA I/O's.

For more information on quick switches visit www.idt.com/quickswitch

Note – These IOs are available only when using XC2VP40 or XC2VP50 device.

A.9.6 Power Supply

When Board is to be used as PCI add on card then Board can be powered from PC's SMPS. While using the board in standalone mode an external power supply will power the board. Required Vccint (Core) and Vcco voltages are generated using two DC-DC converters - PTH05010W. (Xilinx recommends these parts). Vccaux and Rocket IO supplies; 2.5 V are generated using LDO (LT1963). 5VA & 3.3VA required by the analog section (ADC and DAC) is generated using regulators LM317.

A.9.7 User LEDs

8 LEDs are provided on board, which can be used by user to monitor signals from his design.

A.9.8 Reset Switch

Can be used by user as a manual Reset input while verifying his designs.

APPENDIX B
DESIGN FLOW ON FPGA

B.1 INTRODUCTION

A CAD system is a set of a number of CAD tools needed for a complete digital design. A CAD system typically includes tools for the following tasks: design entry, synthesis and optimization, simulation, and physical design. Following flow-chart explains the steps and functions available with a typical CAD system named as ISE used for the programming of XILINX FPGAs [14].

B.2 SPECIFICATION

The starting point in the process of designing a logic circuit is the conception of the functions, which are desired by the application, and formulation of its general structure. This step is manual step and the designer is very responsible for this because this step requires design experience and intuition. In addition, other performance and physical implementation based constraints, which are desired to be satisfied by the design, should be decided in this step only.

B.3 DESIGN ENTRY

Design entry is the first step of design process within a CAD system. In this step, the formulated design is entered to the CAD system. There may be more than one ways of design entry for a typical CAD system. Generally, three different ways of design entries are seen: Schematic Capture, Truth Table, and Hardware Description Languages (HDL). The Hardware Description Languages are the most popular due to their strong modeling power for complex design. A HDL is similar to a typical computer programming language except that an HDL is used to describe the hardware rather than a program to be executed on a computer. Many commercial HDLs are available.

Two HDLs are IEEE standards: VHDL (Very High Speed Integrated Circuit Hardware Description Language) and Verilog HDL. Both languages are in widespread use in the industry. However, in current work VHDL is being used because it is more popular than Verilog due to some additional advantages. The main advantage of using VHDL is the design portability. It is because most of the companies, which are offering digital hardware solutions, are supporting VHDL. Hence, by using a standard language the designer can focus on the required functionality of the desired circuit without being overly concerned about the details of the technology that will be used for implementation.

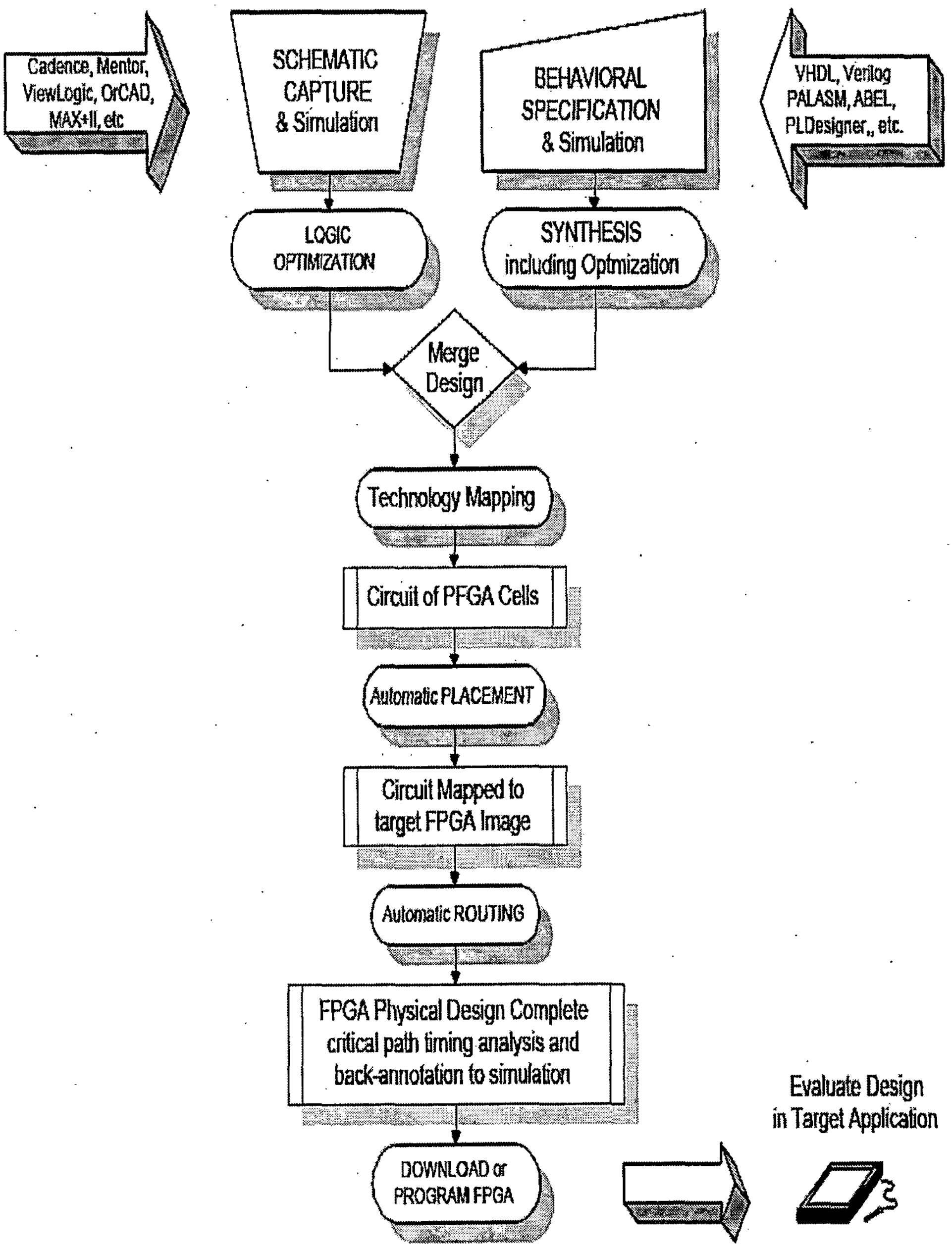

Cadence, Mentor, ViewLogic, OrCAD, MAX+II, etc
SCHEMATIC CAPTURE & Simulation
BEHAVIORAL SPECIFICATION & Simulation
VHDL, Verilog PALASM, ABEL, PLDesigner,, etc.
LOGIC OPTIMIZATION
SYNTHESIS including Optmization
Merge Design
Technology Mapping
Circuit of PFGA Cells
Automatic PLACEMENT
Circuit Mapped to target FPGA Image
Automatic ROUTING
FPGA Physical Design Complete critical path timing analysis and back-annotation to simulation
Evaluate Design in Target Application
DOWNLOAD or PROGRAM FPGA

Figure B.1 : Design flow for FPGA in a typical CAD system

B.4 SYNTHESIS

Synthesis is the process of generating a logic circuit from a truth table. Synthesis CAD tools perform this process automatically. However, the synthesis tools also handle many other tasks. The process of translating, or compiling, VHDL code into a network of logic gates is part of synthesis. Figure B.1 shows a simple design flow inside a typical CAD system for FPGAs. A netlist is generated using the VHDL code and a logic synthesis tool using Xilinx ISE 7.1i EDA tool, synthesis report gives idea about possible shortfalls in the generated RTL model.

B.5 PLACE AND ROUTE

The place process decides the best location of the cells in a block based on the logic and desired performance. The route process makes the connections between the cells and the blocks. The synthesis tool does automatic place and route after generating netlist.

B.6 CONFIGURATION

This is done by loading the configuration data into the internal memory. Synthesis tool generates a bit stream file after placing and routing, which is then downloaded in FPGA.

B.7 VERIFICATION

At each step of the design process, current architecture using software simulation using ISE simulator software package. Each instruction is simulated and its impact is studied to make any changes in design to optimize the performance. Any third party simulator can not be used in the current case as the design uses Xilinx IP cores, whose simulation models are not available with any third party software.