

P- 3463



**VLSI IMPLEMENTATION OF AC BITPLANE ENCODER OF
CCSDS COMPRESSION STANDARD**

By

V.SRIRAMYA

Reg. No. 0920106016

of

KUMARAGURU COLLEGE OF TECHNOLOGY

(An Autonomous Institution affiliated to Anna University of Technology, Coimbatore)

COIMBATORE – 641 049

A PROJECT REPORT

Submitted to the

**FACULTY OF ELECTRONICS AND COMMUNICATION
ENGINEERING**

In partial fulfillment of the requirements

for the award of the degree

of

MASTER OF ENGINEERING

IN

APPLIED ELECTRONICS

APRIL 2011

भारत सरकार
अन्तरिक्ष विभाग
इसरो उपग्रह केन्द्र
पोस्ट बॉक्स नं. 1795, हवाई पत्तन मार्ग
विमानपुरा डाक, बेंगलूरु - 560 017. भारत
दूरभाष :
फैक्स :



Government of India
Department of Space
ISRO Satellite Centre
Post Box. No.1795, Airport Road, Vimanapura Post
Bangalore - 560 017. India
Telephone :
Fax :

CERTIFICATE

This is to certify that the project work entitled “**VLSI Implementation of AC Bitplane Encoder of CCSDS Compression Standard**” is carried out by **Ms. V.Sriramya (Reg No: 0920106016)** a bonafide student of Kumaraguru College of Technology, Coimbatore in partial fulfillment for the award of “**Master of Engineering in Applied Electronics**” of an autonomous institution affiliated to Anna University, Coimbatore during the year 2009-2011. It is certified that she has completed the project successfully. She was sincere during the project period.

Lakshmi
(LAKSHMI.A)
Sci/Engr, 'SD',
Project Guide,
Data Handling Division,
ISRO Satellite Centre,
BANGALORE.

M.V.
(M.VANITHA)
Sci/Engr, 'SG',
Division Head,
Data Handling Division,
ISRO Satellite Centre,
BANGALORE

ACKNOWLEDGEMENT

The satisfaction that accompany the successful completion of any task would be incomplete without mentioning the people who made it possible, whose constant guidance and encouragement crowned my efforts with success. In this regards, i am fortunate to express my heartfelt thanks to **Padmabhusan Arutselvar Dr.N.Mahalingam B.Sc.,F.I.E.,Chairman** and **Dr.B.K.Krishnaraj Vanavarayar B.Com,B.L.,** Co-Chairman, Kumaraguru College of Technology for providing necessary facilities throughout the course.

I am thankful to **Dr.Ramachandran, Principal,** Kumaraguru College of Technology, Coimbatore for giving me the necessary encouragement and guidance.

I express my sincere gratitude to **Dr.Rajeswari Mariappan, Ph.D.,HOD,** Department of ECE, for giving me the necessary encouragement and guidance.

I am highly grateful to my guide **Mrs.A. Amsaveni M.E.,(Ph.d),** Assistant. Professor(SRG), Department of ECE, Kumaraguru College of Technology, Coimbatore , for providing guidance and encouraging me throughout the course of this project.

My heartfelt thanks to **Ms..R.Latha M.E., (Ph.d),** Associate Professor, Project co ordinator, for her contribution and innovative ideas at various stages of the project to successfully complete this work.

I express my profound gratitude to **Mrs.Vanitha,** Group Director, Digital Systems Group, Data Handling Division ,ISRO Satellite Center. Bangalore, who offered me an opportunity to take up this project..

I am highly grateful to my guide, **Mrs.A.Lakshmi,** Scientist'D', Data handling Division, Digital Systems Group, ISAC, for guiding me throughout the course of this project work.

I express my profound gratitude to **Mrs.V.Kamalapriya ,**Scientist 'D', ISRO Satellite Center, Bangalore, who offered me an opportunity to take up this project work.

Last, but not the least, I would like to express my gratitude to my family members, friends and to all my staff members of Electronics and Communication Engineering department for their encouragement and support throughout the course of this project.

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	i
	LIST OF FIGURES	vii
	LIST OF TABLES	viii
	ORGANIZATION PROFILE	xii
1	INTRODUCTION	1
	1.1 Prelude	1
	1.2 Background	1
	1.2.1 About CCSDS	1
	1.2.2 Requirements	3
	1.3 Objectives of the project	4
	1.4 Problem definition	4
	1.4.1 Project plan	4
	1.5 Literature Survey	5
	1.6 Organization of thesis	6
2	DATA COMPRESSION	8
	2.1 Overview of compression	8
	2.1.1 Data compression model	9
	2.1.2 Constraints and purpose of compression	10
	2.1.3 Advantages and Disadvantages of compression	11
	2.2 Image compression System	11
	2.2.1 Elementary concepts	12

	2.2.1.1 Image types	12
	2.2.1.2 Entropy	13
	2.2.2 Principles behind image compression	14
	2.2.3 Methods of image compression	15
	2.2.3.1 Lossless compression	15
	2.2.3.2 Lossy compression	16
	2.3 Requirements of image compression in space missions	16
	2.4 Recommendations and standards	17
3	ELEMENTS OF IMAGE DATA COMPRESSION	18
	3.1 Introduction	18
	3.2 Decorrelation	18
	3.3 Quantization	21
	3.4 Entropy coding	21
	3.4.1 Types of entropy coding	22
	3.4.1.1 CCSDS Rice algorithm	23
4	OVERVIEW OF IMAGE COMPRESSION ALGORITHMS	
	4.1 Introduction	24
	4.2 JPEG	24
	4.3 JPEG 2000	25
	4.3.1 Pre-processing	27
	4.3.2 Forward Transform	27
	4.3.3 Quantization	27
	4.3.4 Entropy Coding	27
	4.3.5 Bit stream	28
	4.3.6 Advantages and disadvantages of JPEG2000 compression scheme	28
	4.4 CCSDS recommended compression scheme	29

5	CCSDS RECOMMENDED IMAGE COMPRESSION ALGORITHM	30
	5.1 Overview	31
	5.1.1 General	31
	5.1.2 Data delivery	31
	5.2 The Discrete Wavelet Transforms(DWT)	32
	5.2.1 Architecture	32
	5.2.2 DWT Coefficient Structure	33
	5.3 Bitplane Encoder	33
	5.3.1 Architecture	40
	5.3.2 Specifying AC bit depth in each block	43
	5.3.3 Bitplane coding	44
	5.3.3.1 Overview of coding stages	45
	5.3.3.2 Coding stages 1-3	46
	5.3.3.3 Mapping word to symbols	48
	5.3.3.4 Entropy coding the symbols	50
	5.3.4 Stage 4 coding	52
	5.4 Generation of transition words of a block bitplanewise - an example	53
6	SERIAL ARCHITECTURE	57
	6.1 Proposed Serial Architecture	57
7	RESULTS AND CONCLUSION	
	7.1 Simulation Results	59
	7.1.1 Coefficient Counter, Block Counter and Gaggle Counter	59
	7.1.2 Address Generation and Gaggle Memory	60
	7.1.3 Input Data Latch of Parent, Children and Grandchildren Coefficients	61
	7.1.4 Transition Words of tran B, tran D, tran G, tran H	61

7.1.5 Pre-processing of tran B, tran D,tran G,tranH	62
7.1.6 Symbol Mapper	63
7.1.7 MinimumCodeSelection	64
7.2 Synthesis Report	64
8 CONCLUSION AND FUTURE SCOPE	69
8.1 Conclusion	69
8.2 Future Scope	69
REFERENCES	70
APPENDIX A	71
GLOSSARY	

LIST OF FIGURES

FIGURE NO	CAPTION	PAGE NO
2.1	Data compression model	9
3.1	CCSDS Rice Algorithm	22
4.1	DCT –Based JPEG Encoder	25
4.2	Block Diagram of JPEG 2000 Encoder	27
4.3	Entropy Coding Unit of JPEG 2000 Engine	29
5.1	General Block Diagram of CCSDS Compressor	31
5.2	Program and Data Flow of DWT module	33
5.3	2-D DWT (one level)	34
5.4	Three Level 2-D DWT Decomposition of image	35
5.5	Schematic of Wavelet – Transformed Image With 64 shaded pixels composing a single block	36
5.6	Organization of 8*8 Coefficients	37
5.8	Program and Data Flow of BPE	39
5.9	Schematic of Wavelet Transformed Image	40
5.10	Structure of single coded segment with all Bitplanes	42
5.12	Coded Bitplane Structure for a Segment	45
5.13	Block- 8*8 DWT coefficients	53
6.1	Proposed Serial Architecture	57
7.1	Coefficient counter waveform	59
7.2	Block counter and Gaggle counter waveform	60
7.3	Address Generation and Gaggle Memory	60
7.4	Input Data Latch waveforms	61
7.5	Transition words waveforms	62
7.6	Preprocessing of tran D waveforms	62
7.7	Symbol Mapper	63
7.8	Minimum code option waveform	64

LIST OF TABLES

TABLE NO	CAPTION	PAGE NO
1.1	Compressor Requirements of CCSDS working group	3
5.1	Integer Mapping of 2 bit words	49
5.2	Integer Mapping of 3 bit words	49
5.3	Integer Mapping of 4 bit words	49
5.4	Variable Length Code Option for 2 bit words	50
5.5	Variable Length Code Option for 3 bit words	51
5.6	Variable Length Code Option for 4 bit words	51
5.7	Identifying Variable Length Code Option	52
5.8	Type s_b sign s_b of Grandchildren coefficient 3 rd BP	54
5.9	Children coefficients 2 nd BP	55
5.10	Grandchildren coefficients 2 nd BP	55
5.11	AC refinement of 2 nd Bitplane	56
5.12	AC refinement LSB BP	56

ACRONYMS

BPE	Bit Plane Encoder
BPP	Bits Per Pixel
CCD	Charge Coupled Device
CCSDS	Consultative Committee for Space Data Systems
CNES	Centre National d' Etudes Spatiales
DCT	Discrete Cosine Transform
DECU	Development and Educational Communication Unit
DFT	Discrete Fourier Transform
DOS	Department Of Space
DSG	Digital Systems Group
DWT	Discrete Wavelet Transform
EC	Entropy Coding
ESA	European Space Agency
FFT	Fast Fourier Transform
GSLV	Geo-stationary Satellite Launch Vehicle
HSV	Human Visual System
IDE	Integrated Development Environment
IEEE	Institute of Electrical and Electronics Engineers

ISU	ISRO Inertial Systems Unit
INCOSPAR	Indian National Committee for Space Research
INSAT	Indian National Satellite
ISAC	ISRO Satellite Centre
ISRO	Indian Space Research Organization
ISTRAC	ISRO Telemetry Tracking and Command Centre
JPEG	Join Photographic Expert Group
JPEG-LS	Joint Photographic Expert Group
JPL	Join Propulsion Laboratory
LEOS	Laboratory for Electro Optic Sensors
LOCO	Low Complexity Lossless Compression
LOT	Lapped Orthogonal Transforms
LPSC	Liquid Propulsion Systems Centre
MCF	Master Control Facility
MPEG	Motion Pictures Expert Group
MSE	Mean Square Error
NASA	National Aeronautics and Space Administration
NESAC	North Eastern Space Applications Centre
NMRF	National Mesosphere stratosphere troposphere Radar Facility

NRSA	National Remote Sensing Agency
PLD	Programmable Logic Device
PRL	Physical Research Laboratory
PSLV	Polar Satellite Launch Vehicle
PSNR	Peak Signal to Noise Ratio
ROI	Region of Interest
SAC	Satellite Application Centre
SDSC	Satish Dhawan Space Centre
STFT	Short Time Fourier Transform
VHDL	VHSIC Hardware description Language
VHSIC	Very High Speed Integrated Chip
VSSC	Vikram Sarabhai Space Centre

ORGANIZATION PROFILE

The Indian Space Program was driven by the vision of Dr. Vikram Sarabhai, considered as the Father of the Indian Space Program. His Vision is quoted below:

“There are some who question the relevance of space activities in a developing nation. To us, there is no ambiguity of purpose. We do not have the fantasy of competing with the economically advanced nations in the exploration of the moon or the planets or manned space-flight. But we are convinced that if we are to play a meaningful role nationally and in the community of nations we must be second to none in the application of advanced technologies to the real problems of man and society.”

VISION

The Department of Space has the primary objective of promoting development and application of space science and technology to assist in all-round development of the nation.

OVERVIEW OF THE ORGANIZATION:

Indian Space Research Organization (ISRO) is one of the prestigious organizations of our nation. The Indian Government established the Department of space in 1972 to promote development and application of space science and technology for socio – economic benefits. Indian Space Research Organization is primary agency under Department of Space, for executing Space programs. The space services needed by the country include telecommunications, nationwide telecast, resource survey and management, environment monitoring, meteorological service and other such activities. These services are implemented through development of operational spacecraft with emphasis on self-reliance.

The Department of Space (DOS) is responsible o the extension of the activity in the country in space science though ISRO. The DOS/ISRO head quarters located at Bangalore providing overall directions to the technical, scientific and administrative functions of the other centers and units given below:

1) Vikram Sarabai Space Center (VSSC), Thiruvananthapuram

It is the lead center for launch vehicle development and it pioneers in rocket research and in planning and execution of launch vehicle development projects of ISRO. It is engaged in design and development of launch vehicle technology for launching satellites.

2) ISRO Satellite Center (ISAC), Bangalore

ISAC is entrusted with the prime responsibility of implementing indigenous spacecraft projects for various scientific technological and application missions. It is responsible for the design, fabrication, testing and management of satellite systems for scientific, technological and application missions.

3) Sriharikota Range (SHAR), SIHARKOTA

It is the main launch center of ISRO, fully equipped with the sophisticated launch facility. It also undertakes large scale processing and ground testing of solid propellant rocket motors.

4) Space Application Center (SAC), Ahmedabad

It is the research center for conceiving and building Systems for practical applications of space technology. The major fields of activity cover Satellite Communications, Remote Sensing, Geology and Meteorology.

5) Liquid Propulsion System Center, Mahendrgiri

Engaged in launch vehicle propulsion systems, engine testing. It undertakes design, development and testing of liquid propulsion systems for ISRO's launch vehicle and satellite programmes.

6) ISRO Telemetry, Tracking and Command Network(ISTRAC),Bangalore

ISTRAC has prime responsibility to provide support for launch vehicle and satellite tracking. It provides telemetry, tracking and command (TTC) support for launch vehicle and satellite missions.

7) **INSAT Master Control Facility(MCF), Hassan**

MCF is charged with responsibility of control and operation of INSAT satellites in orbit, including monitoring the health of the satellites and serving the INSAT system through a network of antennae, earth stations, computers and satellite control facilities. It is responsible for all post-launch operations of INSAT satellite including orbital maneuvers, station keeping and in – orbit operations.

8) **ISRO Inertial Systems Unit(IISU), Thiruvananthapuram**

It carries out design and development of inertial systems for both satellites and launch vehicles.

9) **Development and Communication Unit(DECU), Ahmedabad**

It is involved in the conception, definition, and planning and socio-economic evaluation of space applications programmes.

10) **National Remote Sensing Agency(NRSA), Hyderabad**

NRSA is responsible for acquisition, processing and dissemination of satellite and aerial remote sensing data, training of users scientists in various applications/disciplines and remote sensing application projects/studies in several disciplines for resource mapping/disaster monitoring and essential research and development in all its areas of operation. It facilities for the reception and processing of data from remote sensing satellites. It also undertakes aerial remote sensing.

ISRO Satellite Center (ISAC)

ISRO Satellite Center (ISAC) is one of the major research and development Centers of the Indian Space Research (ISRO). It is the lead center for satellite technology. The primary objectives of ISAC is to develop and operationalize indigenous satellites and use space technology for socio-economic development of the country. It was established in 1972 at Bangalore as the Indian Scientific Satellite Project (ISSP) to build the Nation's first satellite Aryabhata. Today, ISAC has grown into a premier Centre for research and development in satellite technology. Over the years, ISAC has planned and

executed several satellite missions of ISRO. These missions represent a broad spectrum of satellite technology.

ISAC Centre is organized in Mechanical System area, Electronic Systems area, Control and Machine area and Special Project area. The Centre also had independent groups; the groups are Planning and Evaluation Group (PPEG), Systems Reliability Group (SRG), and Computer Information Group (CIG)

Beginning with scientific and application experiments, these have culminated in operational space systems. Having successfully developed and deployed state-of-the-art operational satellites for communication, meteorology and remote sensing, the centre is poised to leapfrog into the next century with advanced versions of satellites like Microwave Remote Sensing Satellites and Direct Broadcasting Satellites.

Digital Systems Group (DSG)

DSG being one of the core groups in ISAC is responsible for the following activities:

- Design, development, testing and delivery of Telecommand and Telemetry systems/packages for satellites.
- Design and development of high frequency Data Handling systems for remote sensing satellites and Solid State Recorder (SSR).
- Technology development works in areas like onboard autonomy, image compression, ASIC and FPGA designs, CCSDS and encryption etc relevant to satellite technology.
- Development of various ground support equipments like encoders, data processing systems, etc.

CHAPTER 1

INTRODUCTION

1.1 PRELUDE

Space agencies all over the world are actively involved in the development of remote sensing instruments with high levels of spatial and spectral resolutions. The products generated by these sensors promise to revolutionize our understanding of climatology, meteorology and land management. As the amount of data generated by these sensors is enormous and as the number of sensors continues to grow, it is clear that the role of data compression will be crucial in this development. At different points in the path from the sensor to the end-user the compression needs will be different and both lossy and lossless compression approaches will be needed. In addition, transmission of a full size, uncompressed image from a satellite takes up quite a significant amount of bandwidth due to the picture's size and complexity. It also takes a significant amount of time to transmit the image down to earth for processing.

1.2 BACKGROUND

In order to thoroughly understand the principles, concepts and need for utilization of compression schemes, background information about various compression techniques had to be gathered for analysis and development. The research in this area includes information on the requirements for image compression, the CCSDS recommendation and subsequent developments.

1.2.1 About CCSDS

Consultative Committee for Space Data Systems (CCSDS) was constituted in 1982 by the major space agencies of the world to provide a forum for discussion of common problems associated with the development and operation of space data systems. It is currently composed of ten member agencies, twenty-two observer agencies, and over 100 industrial associates. The

major space agencies of the world recognize that there are benefits in using standard techniques for handling space data and that, by cooperatively developing these techniques, future data system interoperability will be enhanced. In order to assure that work towards standardization of space-related information technologies provides the maximum benefit for the interested agencies, both individually and collectively, an International Consultative Committee for Space Data Systems is established as a forum for international cooperation in the development of data handling techniques supporting space research, including space science and applications.

Since its establishment, it has been actively involved in developing the following recommendations for data and information-system standards to:

1. Reduce the cost to the various agencies of performing common data functions by eliminating unjustified project-unique design and development.
2. Promote interoperability and cross support among cooperating space agencies to reduce operation costs by sharing facilities.
3. Provide a forum whereby interested agencies may exchange technical information relative to the development or application of standards for space-related information technologies.
4. Identify those common elements of space data systems which, if implemented in a standardized way, will result in significant enhancements in the operation of future cooperative space missions, or in the sharing of mission products.
5. Develop through consensus appropriate Recommendations that will guide the development of agency infrastructure so that interoperability is maximized.
6. Facilitate and promote the use of software and hardware developed under the CCSDS program by all participating agencies. Promote the application of the recommendations within the space mission community.

1.2.2 Requirements

In 1998, Consultative Committee for Space Data Systems (CCSDS) Data Compression working group began effort to establish an image compression recommendation suitable for space-borne applications. The working group agreed that a suitable compressor must meet the requirements listed in Table 1.1. These requirements reflect the envisioned application of real-time hardware compression onboard a spacecraft. The main focus of this project work is on compression of spacecraft on-board data.

Table 1.1: Compressor requirements of CCSDS working group

Sl.NO	Compressor requirements
1	Process both frame and non- frame (push – broom) data.
2	Offer adjustable coded data rate or image quality (up to lossless).
3	Accommodate from 4 – bit to 16 – bit input pixels.
4	Provide real –time processing with space qualified electronics.
5	Require minimal ground operation.
6	Limit the effects of packet loss due to bit errors in transmission channel to a small region of image.

Algorithms considered as candidates for CCSDS image compression recommendation included JPEG2000 as well as algorithms proposed by ESA, NASA and CNES. Candidate algorithms were proposed and performance evaluations were conducted based on both quantitative evaluations of compression effectiveness as well as subjective assessments of image quality.

This project deals with the study, analysis and implementation of image compression algorithm for space applications. This is accomplished by choosing the appropriate image compression algorithm among the various existing image compression algorithms, subjected to requirements such as compression ratio, type of data to be compressed, residual distortion in the reconstructed image, design complexity, etc.

The main focus of CCSDS recommended algorithm are the Discrete Wavelet Transformer and Bit Plane Encoder which compress satellite images to decrease the data volume to be sent back to the earth.

1.3 Objectives of the project

The objectives of the project are as follows:

1. To study and analyze the AC bit plane encoding scheme recommended by CCSDS compression standard.
2. To develop a hardware efficient serial architecture model for AC bitplane encoding .
3. Implementation of the hardware model using VHDL.

1.4 Problem definition

Based on the above said objectives, this project involves the “VLSI Implementation of AC bitplane encoding of CCSDS compression algorithm-A serial architecture”.

In high resolution satellites, it is necessary to compress the images to meet the storage and bandwidth requirement. JPEG 2000 is the state of art compression standard used for such applications. JPEG 2000 provides superior image quality at higher compression ratios. Hardware implementation of JPFG 2000 requires more memory and logic usage because of the complexity of the algorithm. CCSDS [Consultative Committee for Space Data Systems] has introduced a DWT [Discrete Wavelet Transform] based compression system whose complexity is less than JPEG 2000 at least by a factor of 2. The lower complexity of the CCSDS standard supports fast and lower power hardware implementation.

Hardware for bitplane coding based on the algorithm recommended by CCSDS has to be implemented in VHDL language. The serial architecture is targeted for efficient implementation in terms of memory reduction and hardware requirements.

1.4.1 Project Plan

To achieve the objectives, the project is split up into different work packages:

1. Identification of the requirements for spacecraft onboard image compressor.

2. Study of existing and upcoming image data compression systems.
3. Detailed study of AC bitplane encoding scheme.
4. To design an hardware efficient serial architecture for AC bitplane encoder.
5. To implement the proposed serial architecture using VHDL a hardware description language.

1.5 Literature Survey

Image compression has been studied since 1950's as it has proved to be advantageous for many applications. Since more and more digital images are used, the study of image compression becomes more essential because uncompressed digital images require considerable storage capacity and transmission bandwidth. Despite rapid progress in mass storage density, processor speed and digital communication system performances, the demand for data storage and data transmission bandwidth continues to outweigh the capabilities of available technologies.

Following references were reviewed in order to obtain details on the general concepts of compression, encoding scheme for digital image compression and corresponding algorithm of this project work:

1. Fundamental concepts play a major role in setting a foundation to understand the theory behind any topic. In order to get a thorough knowledge on the concepts of compression, the book by Rafael C. Gonzalez, Richard E. Woods, "*Digital Image Processing*" ^[1] and S. Jayaraman, S. Esakkirajan, T. Veerakumar, "*Digital Image Processing*", was referred to understand the general concepts of compression. Compression schemes which suited the objectives of this project were referred.
2. CCSDS Recommended standard for Lossless data compression has been included in Lossless Data Compression ^[2] Recommendation for Space Data System Standards.

This report describes a Recommended Standard for a data compression algorithm applied to two-dimensional digital spatial image data from payload instruments and specifies how this compressed data shall be formatted into segments to enable decompression at the receiving end. Source coding for data compression is a method

utilized an data systems to reduce the volume of digital data to achieve benefits in areas including, but not limited to channel,

- i. Reduction of transmission channel bandwidth
- ii. Reduction of the buffering and storage requirement
- iii. Reduction of data transmission time at a given rate

This recommended standard addresses image data compression, which is applicable to a wide range of space-borne digital data, where the requirement is for a scalable data reduction, including the option to use lossy compression, which allows some loss of fidelity in the process of data compression and decompression.

3. CCSDS Informational report for image data compression has been included in Image data compression^[5] report concerning space data system standards. This report presents a summary of the key operational concepts and rationale which underlie the requirements for the CCSDS recommended standard, image data compression^[2]. This report provides a broad overview of the CCSDS image data compression algorithm.
4. “*The New CCSDS Image Compression Recommendation*”, by Pen-Shu Yeh, Philippe Armbruster, Aaron Kiely, Bart Masschelein, Gilles Moury, Christoph Schafer explains the algorithm adopted in the recommendation consists a two dimensional discrete wavelet transform of the image, followed by progressive bit-plane coding of the transformed data. The algorithm can provide both lossless and lossy compression, and allows a user to directly control the compressed data volume or the fidelity with which the wavelet-transformed data can be reconstructed. The algorithm is suitable for both frame-based image data and scan-based sensor data, and has applications for near-earth and deep-space missions.

1.6 Organization of the Thesis

The thesis is organized as follows:

Chapter 2 is a result of a study on image data compression systems in order to obtain insight on the working, features and limitation of these systems. Overview of elementary concepts of

compression is described. Principle and methods of image compression are included. The requirements of image compression for space missions and the corresponding recommendation and standards are listed out. Finally, the performance measurement parameters are described in this chapter.

Chapter 3 gives an overview of the elements of image compression system.

Chapter 4 provides an overview of the available image compression schemes which has been implemented in the past and their disadvantages. More emphasis on JPEG 2000 is done for comparison purpose. This chapter also describes the overview of current recommended scheme for image compression and its superiority over other schemes.

Chapter 5 describes the recommended algorithm in detail and its features. It also describes compression options and parameters that may be adjusted by a user or an implementer of the recommended standard.

Chapter 6 describes about the serial architecture and chapter 7 presents the simulation results of the algorithm recommended by CCSDS.

Conclusion of the work and future scope are described in chapter 8. Finally references are listed out at the end of this document.

CHAPTER 2

DATA COMPRESSION

2.1 Overview of compression

Uncompressed multimedia (graphics, audio and video) data requires considerable storage capacity and transmission bandwidth. Despite rapid progress in mass storage density, processor speeds and digital communication system performance, the demand for data storage capacity and data transmission bandwidth continues to outstrip the capabilities of available technologies. The recent growth of data intensive multimedia-based web applications has not only the need for more efficient ways to encode data and images but have made compression of such signals central to storage and communication technology.

At the present state technology, the only solution is to compress multimedia data before its storage and transmission, and decompress it at the receiver for playback. For example for a compression ratio of 32:1, the space, bandwidth and the transmission time requirements can be reduced by a factor of 32, with acceptable quality. So the study of compression has become an integral part of image processing applications.

Data compression is the art or science of representing information in compact form. These compact representations are created by identifying and using structures that exist in the data. It is used in Joint Photographic Expert Group (JPEG) and Motion Pictures Expert Group (MPEG), which are standards for representing image and video respectively. Data compression algorithms are used in these standards to reduce the number of bits required to represent an image and video respectively. The reason for the need of data compression is that more and more of the information that is generated is in digital form-in the number represented by bytes of data. And the number of bytes required to represent multimedia data can be huge.

According to Lee D Davisson and Robert M Gray, Data Compression is a science and/or art of processing information with at most a tolerable loss of fidelity. The first priority among the designers will be to find an algorithm that can provide highest possible compression. But, the compression ratio cannot be increased beyond a certain point without introducing come

distortion in the reconstructed or decompressed image. For, a given image, this point is defined as the entropy, which is average information expressed in bits per pixel.

The development of data compression algorithms for a variety of data can be divided into two phases;

1. Modeling : Here information about any redundancy that exists in the data is extracted and represented in the form of a model .
2. Coding : A description of the model and a description of how the data differ from the model are encoded, generally using a binary alphabet.

2.1.1 Data compression model

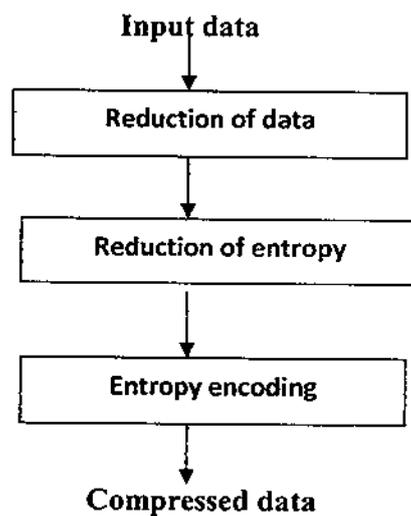


Figure 2. 1 : Data compression model

- i) Reduction of data redundancy can be achieved by transforming the original data from one form to another form. Commonly used methods are Discrete Cosine transform (DCT), Discrete Wavelet transform (DWT) etc.
- ii) Reduction of entropy is done in order to allocate fewer bits for transmission or storage. This is achieved by dropping non significant information in the transformed data.
- iii) The quantized coefficients are losslessly encoded using some entropy scheme to completely represent the quantized data for storage or transmission.

2.1.2 Constraints and Purpose of Compression

In transmission applications, such as broadcast television, space remote sensing via satellite, aircraft, radar, sonar, teleconferencing, computer communications, facsimile transmission, and multimedia systems, space compression techniques are constrained by real time and online considerations that limit the size and complexity of the hardware.

There are three main reasons why present multimedia systems require that data must be compressed:

- 1) Large storage requirements of multimedia data
- 2) Relatively slow storage devices that do not allow plain multimedia data (especially video) in real time , and
- 3) The present network bandwidth, which does not allow real time video data transmission.

The purpose of data compression is to reduce the some part in the data or represent in a different way and to reduce the time for transfer. There are three main types of redundancy that exists in a real time data.

- 1) Coding redundancy (fixed length codes are inefficient)
- 2) Inter pixel redundancy (correlation between pixels)
- 3) Psycho visual redundancy(some information /data are unimportant to the human visual system)

Compression of image data without magnificent loss of visual information is mainly due to three reasons. First, image is contained high degree of spatial redundancy because of correlation between neighboring pixels. Second, they contain some spectral redundancy because of correlation among the color components. Third, they show some degree of psycho visual redundancy due to human beings visual system. From the theoretical view, one should obtain as g high compression as possible based on the redundancy information in the image data. The process of data compression by redundancy reduction is also referred to as source coding. Statistical (spatial) redundancy exists because the pixel values of images usual are not totally random, but represent a degree of gradual changes. Psycho visual redundancy is because the human visual system is insensitive to some spatial frequencies.

2.1.3 Advantages and disadvantages of compression



P. 3463

Advantages of data compression:

1. Compression is useful because it helps reduce the consumption of expensive resources, such as hard disk space and transmission bandwidth.
2. The audience can experience rich-quality signals for audio-visual data representation.
3. Data security can also be greatly enhanced by encrypting the decoding parameters and transmitting them separately from the compressed database files to restrict access of proprietary information.
4. The rate of input - output operations in a computing device can be greatly increased due to shorter representation of data.

Disadvantages of data compression:

1. Extra overhead incurred by encoding and decoding process is one, which discourages its usage in some areas. This extra overhead is usually required to uniquely identify or interpret the compressed data.
2. It generally reduces the reliability of the data records. For example, a single bit error in the compressed code will cause the decoder to misinterpret all subsequent bits, producing incorrect data.
3. Transmission of very sensitive compressed data through a noisy communication channel is risky because the burst errors introduced by the noisy channel can destroy the transmitted data.
4. In many hardware and systems implementations, the extra complexity added by data compression can increase the system's cost and reduce the system's efficiency, especially in the areas of applications that require very low- power VLSI implementation.

2.2 Image compression system

An image data compression system plays a major role in modern digital communication systems. Right from medical imaging, internet applications, Internet applications, remote sensing to multimedia, all of them use compression in order to reduce the number of bits required to be stored and transmission. This saves bandwidth and memory storage capacity.

In the last few decades, several image data compression algorithms have been developed. Most of them have a common building block and can be represented in a general scheme. Wavelet based and advanced encoding techniques allow creating a better and more efficient algorithm, with far more interesting features.

The main objective of the study of image compression algorithms is to check for the feasibility with which these algorithms can be used for onboard compression. The perceived knowledge also helps in correlating the similarities and differences with the JPEG2000 algorithm.

2.2.1 Elementary concepts

An image is a rectangular array of dots, or picture elements, arranged in m rows and n columns. The expression $m*n$ is called the resolution of the image, and the dots are called pixels. The term “resolution” is sometimes also used to indicate the number of pixels per unit length of the image.

2.2.1.1 Image types:

1. A *bi_level* (or monochromatic) image: this is an image where the pixels can have one of two values, normally referred to as black and white. Each pixel in such an image is represented by one bit, making this the simplest type of image.
2. A *gray scale* image: A pixel in such an image can have one of the n values 0 through $n-1$, indicating one of 2^n shades of gray (or shades of some other color). The value of n is normally compatible with a byte size i.e., it is 4, 8, 12, 16, 24, or some other convenient multiple of 4 or of 8. The set of most significant bits of all the pixels is the most – significant bit plane. Thus, a grayscale image has n bit planes.
3. A *continuous-tone* image: This type of image can have many similar colors (or grayscales). When adjacent pixels differ by just one unit, it is hard or even impossible for the eye to distinguish their colors. As a result, such an image may contain areas with colors that seem to vary continuously as the eye moves along the area. A pixel in such an image is represented by either a single large number (in the case of many grayscales) or three components (in the case of a color image). A continuous-tone images normally a

natural image (natural as opposite to artificial) and is obtained by taking a photograph with a digital camera, or by scanning a photograph or a painting. Compression methods for continuous-tone images often do not handle sharp edges very well, so special methods are needed for efficient compression of these images.

4. A *discrete-tone* image: it is also called a graphical image or a synthetic image. This is normally an artificial image. It may have a few colors or many colors, but it does not have the noise and blurring of a natural image. Adjacent pixels in a discrete tone image often are either identical or vary significantly in value. Such an image does not compress well with lossy methods, because the loss of just a few pixels may render a letter illegible, or change a familiar pattern to an unrecognizable one.

2.2.1.2 Entropy

Information theory uses the term entropy as a measure of the amount information generated by the source. The word entropy was borrowed from thermodynamics, and it has a similar meaning. The higher the entropy of a message, the more the information it contains. Mathematically, entropy of a symbol is defined as the negative logarithm of its probability. To determine the information content of a message in bits, we express the entropy using the base 2 algorithm.

The entropy of an entire message is simply the sum of the entropy of all individual symbols. Entropy fits with data compression in its determination of how many bits of information are actually present in a message.

Consider set of 'n' independent events A_i , which are sets of outcomes of some experiment S, such that

$$\cup A_i = S \quad i = 0 \dots (n-1)$$

Where S is the sample space, then the average self-information associated with the random experiment is given by

$$H = \sum_{i=0}^{n-1} P(A_i) i(A_i) = - \sum_{i=0}^{n-1} P(A_i) \log_b (P(A_i))$$

This quantity is called the entropy associated with the experiment. One of the many contributions of Shannon was, is showed that if the experiment is a source that puts out symbols A_i , from a set of A , then the entropy is a measure of the average number of binary symbols needed to code the output of the source. Shannon showed that the best lossless compression scheme can do is to encode the output of a source with an average number of bits equal to the entropy of the source. The coding rate is the average number of bits used to represent a symbol from a source and for a given probability model, the entropy is the lowest rate at which the source can be coded. Because of the fact that the entropy represents a critical point for the maximum compression ratio, data compression scheme as been roughly classified into two basic types, redundancy reduction and entropy reduction. The above classes are represented in literature with different names and they include the following:

- i) Redundancy reduction – reversible, noiseless, lossless
- ii) Entropy reduction – irreversible, lossy.

2.2.2 Principles behind Image compression

A common characteristic of most images is that images is that the neighboring pixels are correlated and therefore contain redundant information. The foremost task then is to find less correlated representation of the image. Two fundamental components of compression are redundancy and irrelevancy reduction. Redundancy reduction aims at removing duplication from the single source (image/video). Irrelevancy reduction omits parts of the signal that will not be noticed by the signal receiver namely Human Visual System (HVS). In general three types of redundancy can be identified

- i. Spatial redundancy or correlation between neighboring pixels
- ii. Spectral redundancy or correlation between different color planes or spectral bands.

- iii. Temporal redundancy or correlation between adjacent frames in sequence of images (in video applications).

Image compression research aims at reducing the number of bits needed to represent an image by removing the spatial and spectral redundancies as much as possible. The information contained in a image must, therefore, be compressed by extracting only visible elements, which are then encoded. The quantity of data involved is thus reduced substantially. The fundamental goal of image compression is to reduce the bit rate for transmission or storage while maintaining an acceptable fidelity or image quality.

The principle of image compression as another aspect. We know from experience that the brightness of neighboring pixels is also correlated. Two adjacent pixels may have different colors. One may be mostly red, and the other may be mostly green. Yet if the red component of the first is bright, the green component of its neighbor will, in most cases, also be bright. This property can be exploited by converting pixel representation from RGB to three other components, one of which is the brightness, and the other two represent color. One such format (or color space) is YC_bC_r , where Y (the “luminance” component) represents the brightness of the pixel, and C_b and C_r components compresses the image while introducing distortions to which the eye is not sensitive. Losing information in the Y component, on the other hand, is very noticeable to the eye.

2.2.3 Methods of Image compression

Since image files not only require large storage but also take up a large portion of bandwidth during transmission, the process of compressing images as become a necessity. Compression can be classified as either lossy or lossless.

2.2.3.1 Lossless Compression

Lossless compression techniques, as the name implies, involve no loss of information. If the data have been losslessly compressed, the original data can be recovered exactly from the compressed data. Lossless compression is generally used for Applications that cannot tolerate any difference between the original and reconstructed data.

Lossless compression is used in medical images, authentic signature and archives imaging but the achievable compression ratio is generally low, from 1:5:1 to 2:1. Text

compression, compression of radiological images, data obtained from satellites are important applications of lossless compression.

The primary goal of lossless compression is to minimize the number of bits required to represent the original image samples without any loss of information or with very small loss of information. Text files containing computer programs may become worthless if even one bit gets modified. Such files should be compressed only by a lossless compression method.

To evaluate lossless compression performance, compressed bit rate in bits/pixel achieved by the lossless compressors on the test images are calculated.

2.2.3.2 Lossy compression

Lossy compression techniques involve some loss of information, and data that have been compressed using lossy techniques generally cannot be recovered or reconstructed exactly. In return for accepting this distortion in the reconstruction, one generally obtains much higher compression ratios than is possible with lossless compression.

Lossy compression can achieve compression ratio as high as 50:1. The secret lies in the fact that the pixel values can be represented in values approximate to the original pixel values. Since human eyes cannot easily distinguish this fine difference, it embodies the tradeoff between accuracy and space.

Lossy algorithms are used in an abstract sense, which does not mean loss of random pixels, but instead means loss of a quantity such as a frequency component, or perhaps loss of noise. Such algorithms make sense especially in compressing images, movies, or sounds.

To evaluate lossy compression performance, the PSNR and MSE metrics are calculated at several different compressed bit rates on the test images.

2.3 Requirements of image compression in space mission:

An image data compression algorithm should be able to:

1. Work with both push broom (non-frame based) data as well as 2D imagery data. Hence it can be used for various types of imaging instruments that allow both modes of operation.
2. Offer an adjustable data rate or image quality based on the requirements.
3. Store and transmit images in several pieces, allowing progressive transmission of images to the ground.

4. Compress autonomously with minimal ground operation, because it is not always possible to interact instantaneously with space segment. The algorithm should work with parameters that adapt to the statistical variations of the image/source data.
5. Provide both lossless and lossy compression options.

2.4 Recommendations and standards

In space industry, there are several organizations which provide recommendations and set standards for space applications. One such organization is CCSDS which has recommended a standard for space data compression system and it has been documented in image data compression [1], Recommendation for Space Data System Standards, CCSDS 122.0-B-1, Blue Book and Image Data Compression[2], a report concerning space data system standards, CCSDS 120.1-G-1, Green Book. The present recommended standard for image compression has the following features:

1. It specifically targets high-rate instruments used on board of spacecraft.
2. A trade-off has been performed between compression performance and complexity with particular emphasis on spacecraft applications.
3. The lower complexity of this recommendation supports fast and low-power hardware implementation.
4. It has a limited set of options, supporting its successful application without in-depth algorithm knowledge.
5. This Recommended Standard supports both frame-based input formats produced by CCD arrays (called image frames) and strip-based input formats produced by push-broom type sensors.
6. An image pixel dynamic range of up to 16 bits is supported.
7. The algorithm specified supports a memory-effective implementation of the compression procedure which does not require large intermediate frames for buffering.

CHAPTER 3

ELEMENTS OF IMAGE DATA COMPRESSION

3.1 Introduction

A typical image compression system is generally comprised of three parts: transformer, quantizer and coder.

Transformer sets a one-to-one relationship between the transformed image and the input image. It reduces redundant information and produces an image representation more amenable to efficient compression than the 'raw material' of input image data.

Quantizer generates a limited number of symbols suitable for representing the compressed image as input. Fundamentally, it is a lossy step, and it is the key difference between a lossy compression method and a lossless one. This step does not exist in a lossless approach. The reason why it is an irreversible process lies in the fact it uses a many-to-one mapping. There are two types of quantization: scalar quantization and vector quantization. The former quantizes the image data on a pixel-by-pixel basis, whereas the later quantizes on a vector basis.

Coder encodes the transformed coefficients into a binary bit stream. The coder can adopt a fix-length or a variable length(entropy)coding scheme. The former assigns all symbols equal length codes, while the latter assigns shorter codes to symbols having more probability of occurrence.

3.2 Decorrelation

An image is represented as a two-dimensional array of coefficients, each coefficient representing the brightness level of that pixel. When looking from a higher perspective, one cannot differentiate between coefficients as more important ones, and lesser important ones. Most natural images have smooth color variations, with the fine details being represented as sharp edges in between the smooth variations. Technically, the smooth variations in color can be termed as low frequency variations and the sharp variations as high frequency variations.

The low frequency components (smooth variations) constitute the base of an image, and the high frequency components (the edges which give the detail) add upon them to refine the image,

thereby giving a detailed image. Hence, the smooth variations are demanding more importance than the details.

Separating the smooth variations and details of the image can be done in many ways. One such way is the decomposition of the image using a Discrete Wavelet Transform (DWT). The past decade saw the birth and rapid advancement of a young branch of applied mathematics, wavelet theory. The many promising and potential advantages of wavelet theory lead to the development of wavelet theory based image processing compression techniques which have improved performance over existing standards like JPEG2000 and has been included as the transformation technique in the current recommendation also.

Over the years, a variety of linear transforms have been developed which include Discrete Fourier Transform(DFT), Discrete Cosine Transform(DCT), Discrete Wavelet Transform(DWT), and many more, each with its own advantages and disadvantages.

The transform of a signal is just another form of representing the signal. It does not change the information content present in the signal. The Wavelet Transform provides a time-frequency representation of the signal. It was developed to overcome the short coming of the Short Time Fourier Transform (STFT), which can also be used to analyze non-stationary signals.

Some of the advantages of DWT which has made it the suitable candidate for CCSDS recommended compression scheme are:

- i. Wavelet Transform uses multi-resolution technique by which different frequencies are analyzed with different resolutions. While STFT gives a constant resolution at all frequencies
- ii. Since there is no need to block the input image and its basis functions have variable length, wavelet coding schemes at higher compression avoid blocking artifacts. In other words, the overlapping nature of the wavelet transform alleviates blocking artifacts.
- iii. Wavelet-based coding is more robust transmission and decoding errors, and also facilitates progressive transmission of images. In addition, they are better matched to the HVS characteristics.
- iv. Because of their inherent multi resolution nature, wavelet coding schemes are especially suitable for applications where scalability and tolerable degradation are important.

Specific applications include DCT data quantization in JPEG and DWT data quantization in JPEG 2000.

If we consider a digital image, quantization is done by dividing each component coefficient in the frequency domain by a constant for that component and then rounding to the nearest integer. As a result of this, it is typically the case that many of the high frequency components are rounded to zero and many of the rest become small positive or negative number, which take fewer bits to store. As a consequence, the precision is lower.

3.4 Entropy Coding

In information theory, entropy coding is a lossless data compression scheme that is independent of the characteristics of the medium. One of the main characteristics of the entropy coding is to assign codes to symbol with code length of each codeword proportional to the negative logarithm of the probability.

After the data has been quantized into a finite set of values, it can be encoded using an Entropy Coder to give additional compression. By entropy, mean the amount of information present in the data, and an entropy coder encodes the given set of symbols with the minimum number of bits required to represent them.

The process of entropy coding (EC) can be spilt in two parts: modeling and coding. Modeling assigns probabilities to the symbols, and coding produces a bit sequence from these probabilities.

Entropy coding creates and assigns a unique prefix code to each unique symbol that occurs in the input. These entropy encoders then compress data by replacing each fixed-length input symbol by the corresponding variable-length prefix codeword. The length of each codeword is approximately proportional to the negative logarithm of the probability. Therefore, the most common symbols uses the shortest codes.

3.4.1 Types of Entropy Coding

Two of the most common entropy encoding techniques is Huffman coding and Arithmetic coding. The mostly widely known technique, undoubtedly, is the Huffman algorithm

which generates variable length codes optimal for fixed known source distributions. But most real applications produce source symbol distributions which vary, so the optimality of an individual Huffman code is often insignificant over a narrow range of data entropies. Efforts to remedy this limitation are exemplified by the dynamic Huffman code and rice's universal noiseless coding technique.

The procedure of partitioning a sequence of samples, or sample differences, into smaller blocks, and adaptively selecting one of several Golomb codes for losslessly encoding each partition is referred to as Rice Coding.

The CCSDS indicates the Rice coder as a good lossless data compressor to obtain optimal performance for various space applications. The Rice algorithm is an easily implementable and adaptive scheme that codes data close to the source entropy and can be extended to any entropy range, as desired. The current recommendation has adopted a variation of Rice encoder as its entropy coding technique. So a study of the same has been done in the next section.

3.4.1.1. CCSDS Rice algorithm

The Rice compression algorithm, depicted in Figure 3.1 consists of two separate functional parts: a “pre-processor” block, followed by an “Adaptive Entropy Coder”.

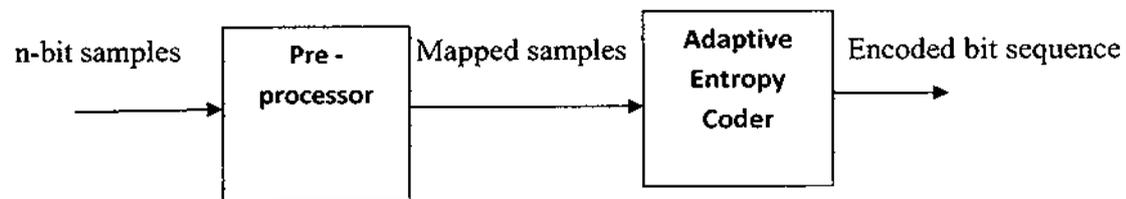


Figure 3.1 CCSDS Rice Algorithm

The input data to the preprocessor, x , is a J -sample block of n -bit samples:

$$X = x_1, x_2, \dots, x_j \quad (3.1)$$

The preprocessor transforms input data into blocks of preprocessed samples Δ , where

$$\Delta = \Delta_1, \Delta_2, \dots, \Delta_j \quad (3.2)$$

The Adaptive Encoder converts preprocessed samples, Δ into an encoded bit sequence.

The preprocessor is a “predictor” block followed by a symbol ‘mapper” block. The function of the predictor is to de-correlate the incoming data stream by simply taking the difference between adjacent data, when we are working with a unit- delay predictor, as recommended by the CCSDS. The function of the mapper is to map all difference values, to a sequence of integer positive symbols. In fact, the second functional block, the Adaptive Entropy Coder, works only with the positive values.

The Adaptive Entropy Coder chooses the best option amongst a set of Code Options, to represent an incoming block of “pre-processed” data samples. A unique binary identifier precedes a coded block to inform the decoder which decoding algorithm to use. The basic code selected is a variable-length code that utilizes Rice’s adaptive coding technique. In Rice coding technique, several algorithms are applied to a block of J consecutive preprocessed samples. The algorithm option that yields the shortest codeword sequence for the current block of data is selected for transmission.

CHAPTER 4

OVERVIEW OF IMAGE COMPRESSION STANDARDS

4.1 Introduction

One question about compression algorithms is how one can judge the quality of one versus another. In the case of lossless compression there are several criteria one can think of, the time to compress, the time to reconstruct, the size of the compressed messages etc. In the case of lossy compression the judgment is further is further complicated since one has to worry about how good the lossy approximation is. There are typically tradeoffs between the amount of compression, the runtime, and the quality of the reconstruction. Depending on the application one might be more important than another and one would want to pick an algorithm appropriately.

In the growing digital world, technologies emerge frequently that give us the ability to compress files at higher and higher ratios which in turn leaves more storage space for files and programs. Digital images are stored in several different formats, or compression styles. One of the most commonly used format of representing digital images today is the Joint Photographic Expert Group (JPEG) standard which is of lossy file format.

This chapter deals with the study that was carried out for selecting a suitable algorithm for compressing the spacecraft on-board image data. In selecting, designing and implementing a compression scheme for onboard system, a number of considerations and trade-off's are to be investigated. Though it cannot be pointed out at length, the major points to be looked into are:

- i. Type of the data to be compressed.
- ii. Requirement of compression ratio.
- iii. Residual distortion in reconstructed image.
- iv. Design complexity.
- v. Fixed rate or variable rate output.
- vi. Sensitivity to transmission channel errors.

4.2 JPEG

The JPEG format make a good real-world example because (a) it is used widely in practice, and (b) it uses many of the compression techniques such as Huffman codes, Arithmetic codes, Residual coding, Runlength coding, Scalar quantization and Transform coding. JPEG is used for still images and is the standard used on the web for photographic images.

JPEG is a lossy compression scheme for color and grey-scale images. It works on full 24-bit colour, and was designed to be used with photographic material and naturalistic artwork.

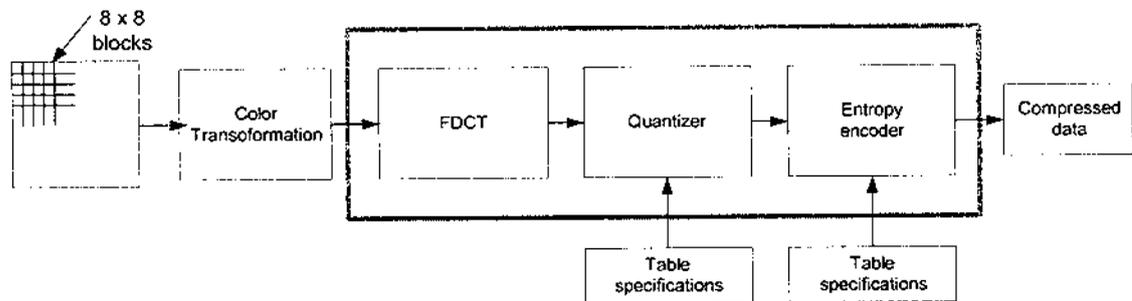


Figure 4.1 DCT-Based JPEG Encoder

Figure 4.1 shows the block diagram of DCT based encoder. The encoding process consists of several steps:

- i. The representation of the colors in the image is converted from RGB to $Y C_B C_R$, consisting of one luma component (Y), representing brightness, and two chroma components, (C_B and C_R), representing colour. This step is sometimes skipped.
- ii. The resolution of the chroma data is reduced, usually by a factor of 2. This reflects the fact that the eye is less sensitive to fine colour details than to fine brightness details.
- iii. The image is split into blocks of 8×8 pixels, and for each block, each of the Y , C_B and C_R data undergoes a Discrete Cosine Transform (DCT).

- iv. The amplitudes of the frequency components are quantized. Human vision is much more sensitive to small variations in colour or brightness variations. Therefore, the magnitudes of the high-frequency components are stored with a lower accuracy than the low-frequency components.
- v. The resulting data for all 8×8 blocks is further compressed with a lossless algorithm, a variant of Huffman encoding.

Although JPEG has its limitations due to it being a “lossy” file format, it has one advantage over others of its kind namely the user has the ability to define the compression ratio of any given picture. In other words the user has the ability to change how much of the quality is lost and how much storage space is saved over the original image. If the user wants to retain almost all of the pixels, then they must sacrifice some storage space that could be saved if a greater compression ratio were to be utilized.

Although JPEG is a widely used file format, especially in the area of online pictures and those taken by digital cameras, it does have some major disadvantages:

- i. The primary downfall to JPEG is that it has a “lossy” format and is therefore prone to many problems.
- ii. JPEG offers an excellent quality at high and mid bit-rates. However, the quality is unacceptable at low bit-rates (e.g. below 0.25 bpp)
- iii. JPEG cannot provide a superior performance at lossless and lossy compression in a single code stream.
- iv. JPEG was optimized for natural images. Its performance on computer generated images and bi-level (text) images are poor.
- v. Another issue with JPEG formatting is that if pixels are related, such as a large number of pixels are all the same colour, then a blurring can occur, or more loss is occurring than originally expected.

4.3 JPEG 2000

JPEG 2000, the newest standard provided by the JPEG still image compression group is a highly sought after algorithm due to its compilation of features, as well as having a “lossy” and lossless version.

JPEG 2000, although having the same name as JPEG, is very different. For instance while JPEG is a “lossy” format, JPEG 2000 is considered both lossless and “lossy”. It is apparent that the JPEG 2000 format retains more quality of the original image and therefore it can be seen why JPEG 2000 is becoming a popular algorithm to implement for image compression

Some of the main features of JPEG 2000 compression scheme include:

- i. The standard provides lossy compression with a superior performance at low bit-rates. It also provides lossless compression with progressive decoding. Applications such as digital libraries/databases and medical imagery can benefit from this feature.
- ii. In this mode, Regions of Interest (ROI’s) can be defined. These ROI’s can be encoded and transmitted with better quality than the rest of the image.
- iii. The standard incorporates a set of error resilient tools to make the bit-stream more robust to transmission errors.

Block diagram of JPEG 2000 encoder is shown in figure 4.2 Each block is described in the coming sections.

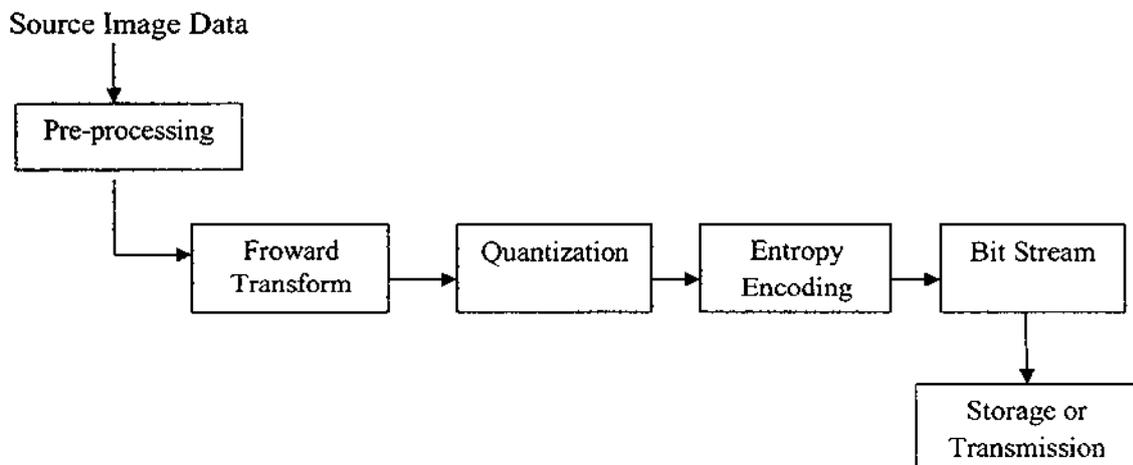


Figure 4.2 Block Diagram of JPEG 2000 Encoder

4.3.1 Pre-processing

Pre-processing includes three steps namely:

- i. Image tiling: this is mainly done in order to keep the computational burden low and for relatively random access in the compressed domain for each image component.
- ii. Colour transformation: It mainly deals with colour conversion for decorrelation i.e. from RGB to YUV or $YCbCr$.

4.3.2 Forward Transform

Discrete Wavelet Transform (DWT) is used to decompose each tile component into different sub-bands. During the wavelet transform, image components are passed recursively through low pass and high pass wavelet filters. This enables an intra component decorrelation that concentrates the image information in a small and much localized area. It also enables the multi-resolution image representation. By itself the wavelet transform does not compress the data. But it restructures the image information so that it is easier to compress.

DWT can be irreversible or reversible.

- i. Irreversible transform: 9-tap/7-tap filter.
- ii. Reversible transform: 5-tap/3-tap filter.

JPEG 2000 supports two filter modes namely:

- i. Convolution based.
- ii. Lifting based.

4.3.3 Quantization

Wavelet transform can be followed by a quantization step in lossy compression. Quantization reduces the bit depth of wavelet coefficients at the expenses of precision. Two different quantization procedures are allowed by the standard: scalar quantization, and Trellis-coded quantization.

4.3.4 Entropy coding

The entropy coding unit is composed of a coefficient bit modeler and the arithmetic coder itself.

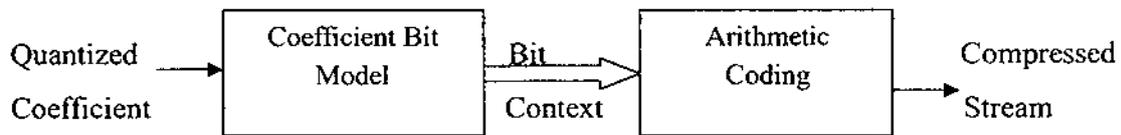


Figure 4.3 Entropy coding unit of JPEG 2000 compression engine

The bit modeler estimates the probability of each possible event at each point in the coding stream. While the arithmetic coder removes the redundancy in the encoding of the data. It assigns short codewords to the more probable events and longer codewords to less probable ones.

4.3.5 Bit stream

The data ordering module embeds all groups of pixels in a succession of packets. These packets along with the additional headers from the final JPEG 2000 bit stream.

4.3.6 Advantages and Disadvantages of JPEG 2000 Compression Scheme

JPEG 2000 has many advantages over the original algorithm:

- i. State-of-the-art low bit-rate compression performance.
- ii. Progressive transmission by quality, resolution, component, or spatial locality.
- iii. Provides both lossy and lossless compression.
- iv. Random (spatial) access to the bit stream.
- v. Region of Interest (ROI) coding by compression.
- vi. Limited memory implementation.
- vii. It is highly resistant to errors that may occur during the transmission of signals over lines or through the air.

Disadvantages of JPEG 2000:

- i. JPEG 2000 is a resources hog. It consumes lot of memory and computation time.
- ii. JPEG 2000 algorithm is more complex compared to the earlier versions, due to which the hardware implementation complexity is high.

4.4 CCSDS Recommended Compression Scheme

CCSDS recommended standard is another wavelet-based image compression that is capable of providing effective lossless and lossy compression. It differs from other compression schemes in several aspects like:

- i. it specially targets use aboard spacecraft.
- ii. a careful trade-off has been performed between compression performance and complexity.
- iii. being less complex, it can be more easily implemented in either hardware or software.
- iv. it has a limited set of options, supporting its successful applications without in-depth knowledge of the algorithm.

CCSDS strip-based compression offers compression effectiveness close to that obtained by JPEG 2000 scan-based compression for all images both for PSNR and MSE metrics. The trade-off in performance is within one dB, and this difference may be smaller than the performance penalty obtained when similar complexity constraints are imposed on JPEG 2000 in a practical hardware implementation.

Taking all these facts and figures into consideration, CCSDS strip based or frame based compression scheme best suites the requirement of space mission since the algorithm involves less options making software implementation less complex. Also it has been found that it offers compression effectiveness close to that of JPEG 2000 performance analysis of algorithm.

CHAPTER 5

CCSDS RECOMMENDED IMAGE COMPRESSION ALGORITHM

5.1 OVERVIEW

5.1.1 GENERAL

This Recommended Standard defines a particular payload image data compression algorithm that has widespread applicability to many types of instruments. The compression technique described in this Recommended Standard can be used to produce both lossy and lossless compression. The compressor consists of two functional parts, depicted in figure 5.1 , a Discrete Wavelet Transform module that performs decorrelation, described in section 5.2 , and a Bit-Plane Encoder which encodes the decorrelated data, described in section 5.3.

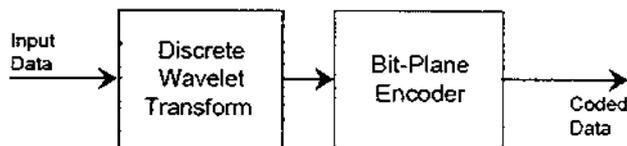


Figure 5.1 General block diagram of CCSDS compressor

This Recommended Standard supports both frame-based input formats produced, for example, by CCD arrays (called image frames) and strip-based input formats produced by push-broom type sensors (called image stripmaps). An image pixel dynamic range of up to 16 bits is supported. The algorithm specified supports a memory-effective implementation of the compression procedure which does not require large intermediate frames for buffering. The compressor relies on a DWT. The Recommended supports two classes of DWT: an integer and a floating point DWT. The integer DWT requires only integer arithmetic, is capable of providing lossless compression, and has lower implementation complexity. The floating point DWT provides improved compression effectiveness at low bit rates, but requires floating point calculations and cannot provide lossless compression.

To limit the effects of data loss that may occur on the communications channel, the wavelet transformed image data are partitioned into *segments*, each loosely corresponding to a different region of the image. Each segment is compressed independently, so that the effects of

data loss or corruption are limited to the affected segment. Partitioning the wavelet-transformed image data into segments also has the benefit of limiting the memory required for some implementations. The segment size can be adjusted to trade the degree of data protection for compression effectiveness; smaller segments provide increased protection against data loss, but tend to reduce the overall compression ratio.

Within each compressed image segment, data are arranged so that earlier portions of the compressed data segment tend to make a larger contribution in overall reconstructed segment fidelity than later portions. This *embedded* data structure allows a user to meet a constraint on compressed segment data volume by truncating the compressed bitstream for a segment at the appropriate point.

5.1.2 DATA DELIVERY

The encoded bitstream corresponding to an image frame or strip map, consists of a single segment or a sequence of segments. Each segment consists of a header followed by a coded data field. A segment can have either fixed length or variable length depending on the operational mode selected.

5.2 THE DISCRETE WAVELET TRANSFORM (DWT)

This Recommended Standard for the decorrelation module makes use of a three-level, two dimensional (2-D), separable Discrete Wavelet Transform (DWT) with nine and seven taps for low- and high-pass filters, respectively. Such a transform is produced by repeated application of a one-dimensional (1-D) DWT. Two specific 1-D wavelets are specified with this Recommended Standard: the 9/7 biorthogonal DWT, referred to as ‘9/7 Float DWT’ or simply ‘Float DWT’, and a non-linear, integer approximation to this transform, referred to as ‘9/7 Integer DWT’ or simply ‘Integer DWT’. While the Float DWT generally exhibits superior compression efficiency in the lossy domain, only the Integer DWT supports strictly lossless compression.

The values output from the 3-level 2-D DWT are converted to appropriate integer values before applying the Bit Plane Encoder. Each integer is represented using a binary word consisting of a

single sign bit along with several magnitude bits. In the case of the Float DWT, the computed wavelet domain values are rounded to the respective nearest integers before applying the BPE.

5.2.1 Architecture

A program and data flow diagram of the transform (DWT) stage of the compressor is shown in figure 5.2. Image data input is shown at left in the diagram. The DWT coefficient buffer at right in the diagram stores wavelet coefficients computed by the DWT stage. The program flow in the diagram produces DWT coefficients for a single segment. Once the coefficients corresponding to a segment have been computed and placed in the buffer, the BPE stage can begin encoding that segment. The BPE stage relies on all of the coefficients in a segment being available simultaneously.

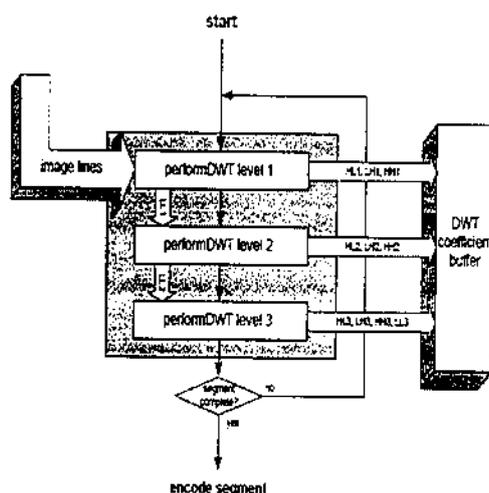


Figure 5.2 Program and data flow of DWT module

5.2.2 DWT coefficient structure

TWO-DIMENSIONAL SINGLE-LEVEL DWT

Image decorrelation is accomplished using a two-dimensional DWT, which is performed by iterated application of the one-dimensional DWT. Viewing the image as a data matrix consisting of rows and columns of signal vectors, a single-level 2-D DWT shall be performed on the image in the following two steps in the following order:

- a) the 1-D DWT shall be performed on each image row, producing a horizontally low pass

and a horizontally high-pass filtered intermediate data array, each half as wide as the original image array, as illustrated in figure 5.3 (b).

b) the 1-D DWT shall be applied to each column of both intermediate data arrays to produce four subbands as shown in figure 5.3 (c).

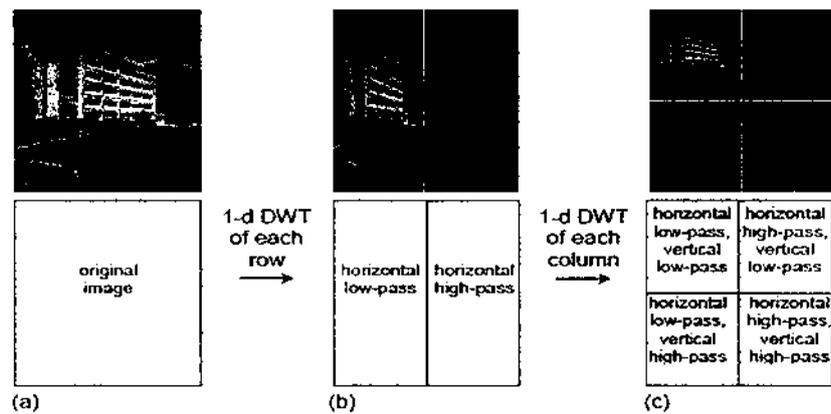


Figure 5.3 - 2-D DWT (One Level)

Each of the four subband data arrays obtained is half as wide and half as tall as the original image array. In illustrations, these subbands are often shown arranged as one array which has the same size as the original image array (see figure 5.3 (c)). Starting at the upper left and proceeding clockwise in figure 5.3 (c), the four subbands are referred to as LL, HL, HH, LH.

MULTI-LEVEL TWO-DIMENSIONAL DWT

To increase compression effectiveness, correlation remaining in the LL subband after the 2-D DWT decomposition is exploited by applying further levels of DWT decomposition to produce a multi-level 2-D DWT.

This Recommended Standard specifies three levels of decomposition. At each level, the 2-D DWT shall be applied to the LL subband produced by the previous level of decomposition.

Figure 5.4 illustrates a three-level 2-D DWT decomposition. At each level of decomposition, the LL subband from the previous level is decomposed, using a 2-D DWT, and is replaced with four

new subbands. Each new subband is half the width and half the height of the LL subband from which it was computed. Each additional level of decomposition thus increases the number of subbands by three but leaves unchanged the total number of DWT coefficients used to represent the image data. Following n levels of 2-D DWT decomposition, the total number of subbands is therefore $3n+1$. Following the recommendation of a three-level decomposition, ten subbands are generated. The subbands are typically shown arranged to form an array of the same dimensions as the original image, as is done in figure 5.4. Subscripts are added to LL, HL, HH, and LH to denote the level of decomposition.

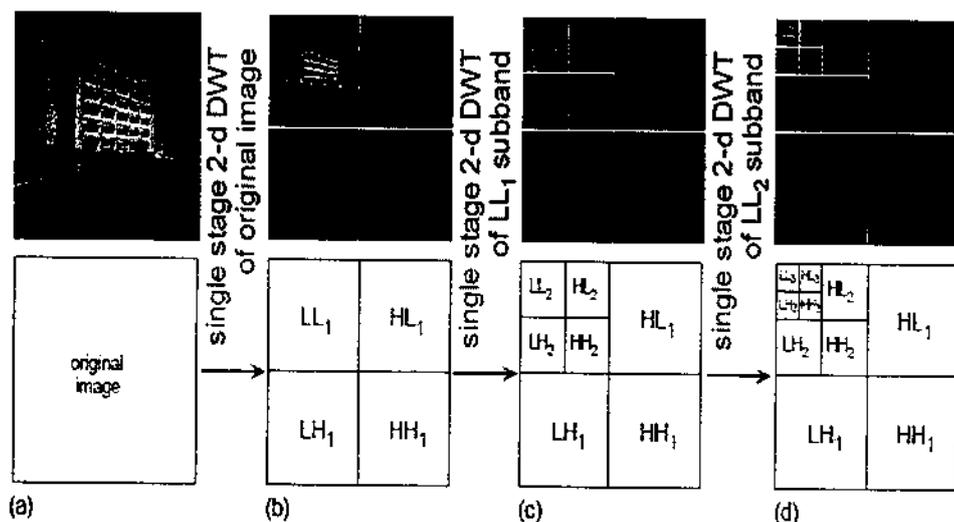


Figure 5.4 : Three-Level 2-D DWT Decomposition of an Image

The BPE processes wavelet coefficients in groups of 64 coefficients referred to as *blocks*. A Block loosely corresponds to a localized region in the original image. A block consists of a single coefficient from the lowest spatial frequency subband, referred to as the *DC coefficient*, and 63 *AC coefficients*, as illustrated in figure 5.5 . This structure is used for jointly encoding information pertaining to groups of coefficients within the block because they exhibit significant statistical correlation.

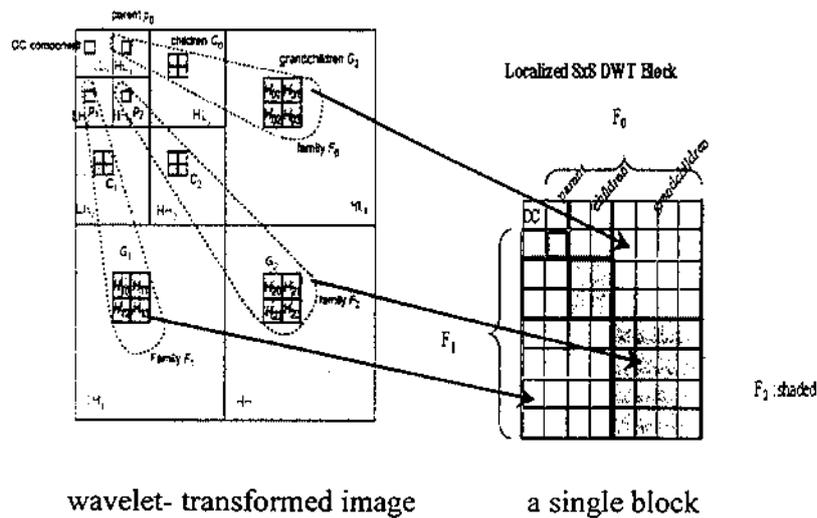


Figure 5.5 Schematic of a Wavelet-Transformed Image with the 64 Shaded Pixels Composing a Single Block

Blocks are processed by the BPE in raster scan order; i.e., rows of blocks are processed from Top to bottom, and proceeding from left to right horizontally within a row. A *segment* is defined as a group of S consecutive blocks. Coding of DWT coefficients proceeds segment by- segment and each segment is coded independently of the others. The value of S may change with each segment.

5.3 THE BITPLANE ENCODER (BPE)

The bit plane encoder takes as input the 8×8 transform domain components and further groups them into three family trees, each has one parent, four children and sixteen grand children as shown in Fig. 5.6. The magnitudes of components are scanned for any most significant bit (MSB) on the scanned bit plane. This bit plane scanning proceeds from the top most bit plane downward, thus inherently accomplishes a quantization of transform coefficients by a factor of 2 in succession. The positional information of those identified components is represented by this

family tree structure and may be further coded for efficiency. This information along with their sign information are shifted to the output bit string from higher bit planes to lower bit planes

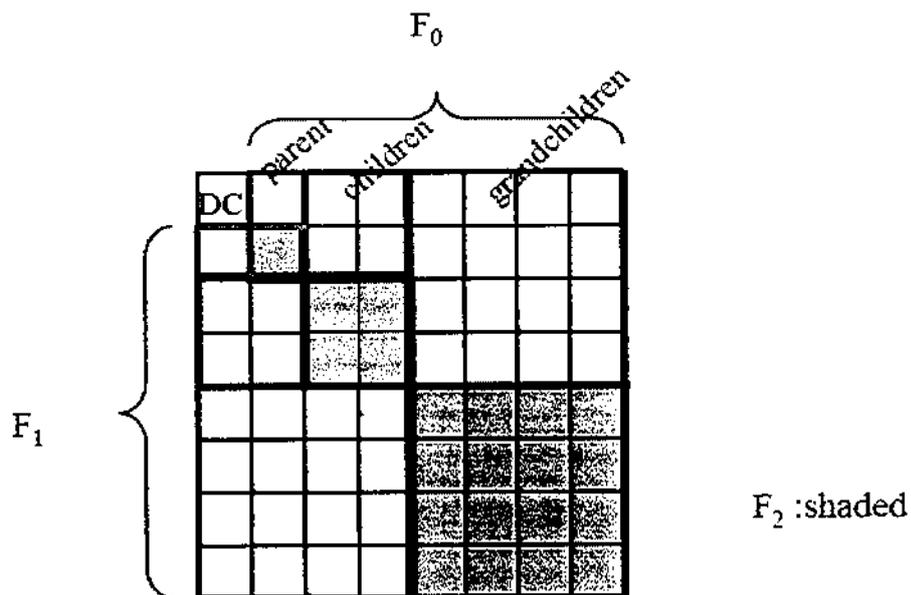


Fig. 5.6 Organization of 8x8 Coefficients

Uniform compression performance over a strip of input data is obtained by processing the same bit plane for all the blocks in a strip of input data to the de-correlator. Fig. 5.6 shows how data from each bit plane are organized and metered out to provide exact compression rate desired by the application. On each bit plane, each pattern class (parent, child or grand child) is processed across all the blocks before progressing to the next pattern class, thus providing sub-bit plane accuracy at user specified compression rate.

Transformed Coefficients in Bit Planes:

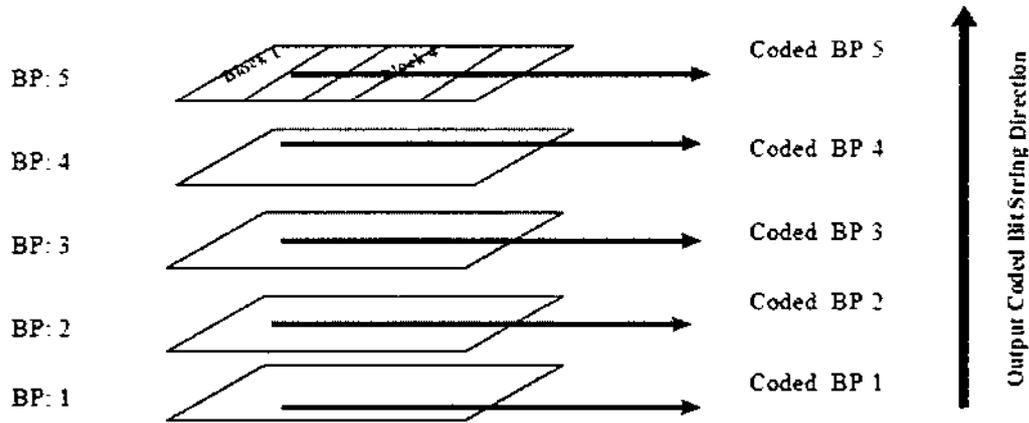


Fig. 5.7 Flow of Coded Data from Bit Planes (BP)

A program and data flow diagram of the BPE stage of the compressor is shown in figure 5.7. The BPE takes DWT coefficient data from the DWT coefficient buffer, encodes coefficient data, and places the encoded output in the compressed data stream. Coding of a segment is done in four steps, each corresponding to a shaded box in the figure. The steps encode the segment header, a quantized representation of DC coefficient information, the bit depths of AC coefficient blocks, and bit planes of AC coefficients. These coding steps are summarized below.

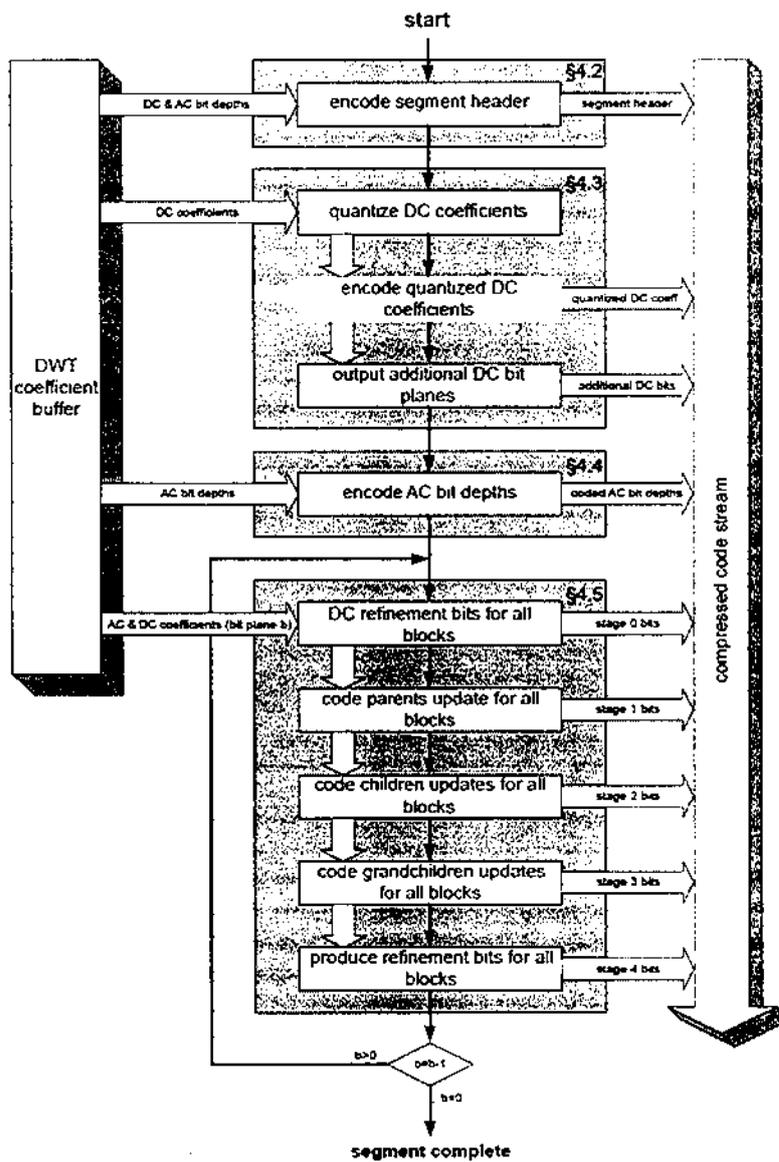


Figure 5.8 : Program and Data Flow of BPE

The program flow in figure 5.8 corresponds to the encoding of DWT coefficient data for a single segment, including complete encoding of all bitplanes.

5.3.1 Overview of BPE

Following the DWT, the wavelet coefficients are either rounded to the nearest integer (when the floating-point transform has been used), or scaled using the weighting factors (when the integer transform has been used). When scaling is performed, one or more of the least-significant bits in many of the subbands are necessarily all zeros. The BPE process takes this into account, i.e., bits that must be all zeros because of the scaling process are not encoded in the BPE. In the description in this section, the coefficient values mentioned refer to values after any scaling or rounding operation. The notation $\text{BitShift}(I)$ is used to indicate the number of least significant bits in each coefficient of the subband I that are necessarily zero as a result of the subband scaling operation. When the 9/7 Float DWT is used, sub-bands are not scaled, thus BitShift equals zero for each subband.

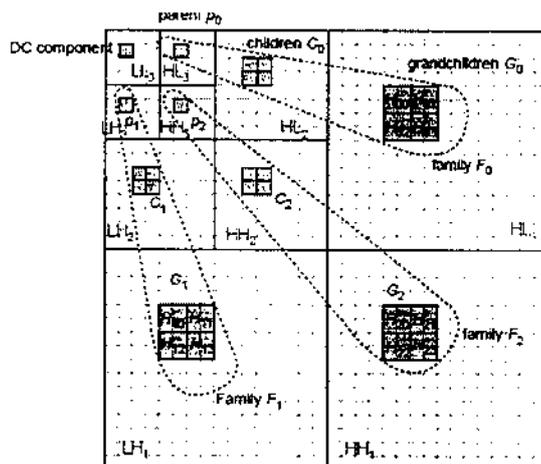


Figure 5.9 -Schematic of Wavelet-Transformed Image

The Bit Plane Encoder (BPE) processes wavelet coefficients in groups of 64 coefficients referred to as *blocks*. An example of a block is illustrated in figure as comprised of shaded pixels. A block loosely corresponds to a localized region in the original image.

Information pertaining to a block of coefficients is jointly encoded by the BPE. A block consists of a single coefficient from the LL3 subband, referred to as the *DC coefficient*, and 63 *AC coefficients*. The AC coefficients in a block are arranged into three *families*, F_0 , F_1 and F_2 . Figure 5.9 illustrates a single block of coefficients and the family structure.

Each family F_i in the block has one *parent* coefficient, p_i , a set C_i of four *children* coefficients, and a set G_i of sixteen *grandchildren* coefficients. The grandchildren in family F_i are further partitioned into groups numbered $j=0,1,2,3$, denoted H_{ij} , as illustrated in figure 5.9.

A *segment* is defined as a group of S consecutive blocks. Coding of DWT coefficients proceeds segment-by-segment and each segment is coded independently of the others. S can be assigned to any value between $16 \leq S \leq 2^{20}$, except for the last segment of an image, for which S can be assigned to any value between $1 \leq S \leq 2^{20}$. The value might be chosen based on the memory available to store the segment. When multiple image frames are transmitted, the coding of each new frame starts with a new segment, i.e., a single segment must not contain coded data from two separate frames.

A segment of blocks is further partitioned into *gaggles*. Each gaggle consists of 16 blocks, except for possibly the last gaggle in a segment, which contains $S \bmod 16$ blocks when S is not a multiple of 16. DC coefficients are represented using two's-complement representation. Let c_m denote the m^{th} DC coefficient in a segment, i.e., the DC coefficient of the m^{th} block in a segment. The number of bits needed to represent c_m in two's-complement representation is given in equation 5.1 & 5.2 for $c_m < 0$ and $c_m \geq 0$ respectively.

$$1 + \text{ceil}(\log_2|c_m|), \text{ if } c_m < 0 \quad (5.1)$$

$$1 + \text{ceil}(\log_2(1+c_m)), \text{ if } c_m \geq 0 \quad (5.2)$$

Within a segment, BitDepthDC is defined as the maximum of this value over all DC coefficients (i.e., all values of m) in the segment. Each DC coefficient in the segment is represented using BitDepthDC bits, in two's-complement representation.

An AC coefficient is represented using the binary representation of the magnitude of the coefficient, along with a bit indicating the sign when the coefficient is nonzero. BitDepthAC_Block $_m$ in equation 13 denotes the maximum number of bits needed to specify the magnitude of any AC coefficient in the m^{th} block.

$$\text{BitDepthAC_Block}_m = \text{ceil}(\log_2(1 + \max x(|x|))) \quad (5.3)$$

where the maximization is over all AC coefficients x in the block.

For each segment, the BPE computes BitDepthAC , which denotes the maximum value of $\text{BitDepthAC_Block}_m$ for the segment is given in equation 5.4,

$$\text{BitDepthAC} = \max_{m=0, \dots, S-1} \text{BitDepthAC_Block}_m \quad (5.4)$$

The BPE successively encodes bit planes of coefficient magnitudes in a segment, inserting AC coefficient sign values at appropriate points in the encoded data stream. *Bit plane b* consists of the b^{th} bit of the two's-complement integer representation of each DC coefficient, and the b^{th} bit of the binary integer representation of the magnitude of each AC coefficient. Here, bit plane index $b=0$ corresponds to the least significant bit. The BPE proceeds from most-significant bit to least significant bit, thus b decreases from one bit plane to the next, beginning with $b = \text{BitDepthAC}-1$, and ending with $b=0$. DWT coefficient resolution effectively improves by a factor of 2 as encoding proceeds from one bit plane to the next.

Figure 5.10 gives an overview of the structure of any single coded segment. Within a segment, header information is encoded. Then quantized DC coefficients from the blocks are encoded. Then AC bit depths are encoded. Then DWT coefficient blocks are encoded, one bit plane at a time, proceeding from the most significant to the least significant bit plane. The coding of a single bit plane is performed in several stages, and the resulting order of code is illustrated in figure 5.11 E.g., parent coefficients are coded in stage 1 for all blocks of the segment before encoding child coefficients in stage 2. The resulting encoded bit stream constitutes an embedded data format that provides progressive transmission.

Segment Header
Initial coding of DC coefficients
Coded AC coefficient bit depths
Coded bit plane $b = \text{BitDepthAC}-1$
Coded bit plane $b = \text{BitDepthAC}-2$
⋮
Coded bit plane $b = 0$

Figure 5.10 Structure of single coded segment with all bit planes.

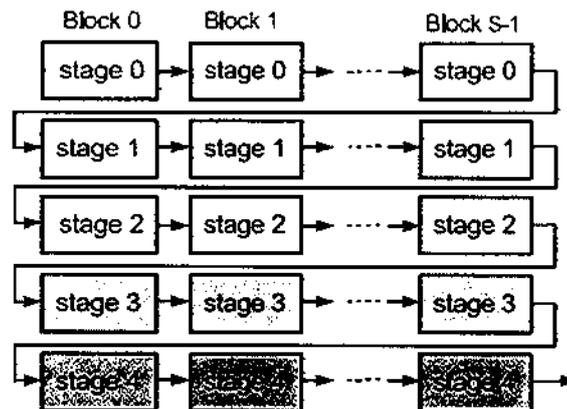


Figure 5.11 Structure of single coded segment within one bitplane

The tradeoff between reconstructed image quality and compressed data volume for each segment is controlled by specifying the maximum number of bytes in each compressed segment, `SegByteLimit`, and a ‘quality’ limit. The quality limit constrains the amount of DWT coefficient information to be encoded, and is specified as a bit plane index and a stopping point within that bit plane. Compressed output for a segment is produced until the byte limit or quality limit is reached, whichever comes first. The encoded bitstream for a segment can be further truncated (or, equivalently, coding can be terminated early) at any point to further reduce the data rate, at the price of reduced image quality for the corresponding segment.

5.3.2 SPECIFYING THE AC BIT DEPTH IN EACH BLOCK

The first step in encoding AC coefficient magnitudes in a segment is to specify the sequence of `BitDepthAC_Blockm` values for the segment. These values shall be coded in one of three ways, depending on the value of `BitDepthAC` that is encoded in the segment header:

- i. If `BitDepthAC` is zero, then the sequence of `BitDepthAC_Blockm` values are zero and thus shall not be coded. In addition, no bit plane coding of the AC coefficients shall be performed since all of the AC coefficients in the segment must be zero.

- ii. If BitDepthAC is one, then each BitDepthAC_Block_m value is either zero or one, and thus each BitDepthAC_Block_m value shall be encoded using a single bit to indicate its value; i.e., the single-bit BitDepthAC_Block_m values shall be concatenated.
- iii. Otherwise, the sequence of BitDepthAC_Block_m values for the segment shall be coded using the same differencing and variable-length coding procedure specified for coding quantized DC values. However, a few parameters differ for the case of coding the BitDepthAC_Block_m values:

N shall equal the number of bits required to represent the magnitude of BitDepthAC for the segment, as in equation 5.5 :

$$N = \text{ceil}(\log_2(1 + \text{BitDepthAC})) \quad (5.5)$$

Since the BitDepthAC_Block_m values are necessarily nonnegative, x_{\min} and x_{\max} shall be redefined as given in equation 5.6 and used to map successive differences to non-negative integers:

$$x_{\min} = 0, x_{\max} = 2N - 1 \quad (5.6)$$

5.3.3 Bit Plane Coding

Coding of a bit plane is performed in *stages* numbered 0-4. The coded bits produced at the stages for each block are interleaved, as illustrated in figure (previous) and figure (5.12). Thus, a coded bit plane first consists of all the stage 0 bits (if any) in the segment, then all of the coded stage 1 bits in the segment, and so on, finishing with all of the encoded stage 4 bits in the segment. This produces an embedded bit string with information from the highest bit plane of all S blocks in the first part of the output bit string followed by information from lower bit planes, and allows progressive decoding of the coded string. This improves image reconstruction quality when the coded bit sequence is truncated.

Stage 0 bits from each block in the segment (if any)
Coded stage 1 from block 0, 1, ..., S-1
Coded stage 2 from block 0, 1, ..., S-1
Coded stage 3 from block 0, 1, ..., S-1
Coded stage 4 from block 0, 1, ..., S-1

Figure 5.12: Coded Bit Plane Structure for a Segment

Note that when the index b of the bit plane being coded is larger than or equal to the AC bit depth of the block, then there **Stage 0** for a block consists of at most a single bit, which is simply the b^{th} most significant bit of the two's-complement representation of the DC coefficient. Note that whenever the bit plane index b satisfies $b \geq q$, this bit value is already known from the DC coefficient information already encoded, and in this case, stage 0 is empty, i.e., no bits are coded in stage 0. Stage 0 is also empty when scaling of the DC coefficient assures that the bit must be zero, i.e., when $b < \text{BitShift}(\text{LL3})$.

The remaining stages (1-4) encode AC coefficient bits. The stage in which bits from AC coefficients in a bit plane are coded depends on the *type* of the AC coefficient at the bitplane, which we now define. At bit plane b , the type of an AC coefficient x , denoted $t_b(x)$, has one of the following values: is nothing to code for the block.

5.3.3.1 Overview of Coding Stages

The coding stages for a block at bit plane b are described in the following paragraphs.

- $t_b(x) = 0$ if $|x| < 2^b$, (x is not due for selection at this bit plane);
- $t_b(x) = 1$ if $2^b \leq |x| < 2^{b+1}$, (x is due for selection at this bit plane);
- $t_b(x) = 2$ if $2^{b+1} \leq |x|$, (x has already been selected at a previous bit plane);
- $t_b(x) = -1$ if $b < \text{BitShift}(\Gamma)$, (x must be zero at this bit plane due to subband scaling).

Here, Γ denotes the subband containing x . Thus, during bit-plane encoding, each AC coefficient typically proceeds from type 0 to 1, to 2, to -1. For a set of coefficients Ψ , we define $t_{\max}(\Psi)$ as the maximum of the coefficient types in Ψ .

An AC coefficient x is said to be *selected* at bit plane b if $t_b(x) = 1$. i.e., the ‘selection’ of a coefficient marks the first bit plane where a non-zero magnitude bit is encoded for the coefficient. Note that $t_b(x) = 1$ if the b^{th} most significant magnitude bit of x is equal to ‘1’ and all more significant magnitude bits of x are equal to ‘0’.

The type of a coefficient determines the stage when coding of a coefficient bit takes place. When an AC coefficient x is of type 0 or 1 (implying $t_{b+1}(x)=0$), the b^{th} most significant magnitude bit of x is coded in stages 1-3. Otherwise, the bit is included, uncompressed, in stage 4 if x is of type 2, or not encoded at all when x is of type -1.

In stages 1-3 of BPE encoding bit plane b , the b^{th} magnitude bit of each AC coefficient x such that $t_{b+1}(x)=0$ is encoded. The b^{th} magnitude bits of the parent coefficients are coded in stage 1, the children in stage 2, and the grandchildren in stage 3. Each of these stages also includes coded bits indicating the sign of each coefficient x for which $t_b(x)=1$. The coding in stages 1-3 makes use of the family structure to group together AC coefficients for entropy coding.

The coding performed in stages 1-3 for a block consists of two parts. First, a sequence of variable length binary words are defined which completely describe the bits to be encoded in these stages. Next, a subset of these words are further entropy coded.

Stage 4 of coding consists of the b^{th} magnitude bit of each AC coefficient x with $t_b(x)=2$. These bits are included in the coded data stream uncompressed.

5.3.3.2 Coding stages 1-3

In addition to the sets C_i, G_i, H_{ij} , defined in 4.1, P is defined as the list of parents in the block:

$$P = \{p_0, p_1, p_2\}.$$

The list of descendants in family i , denoted Di , is defined as

$$D_i = \{C_i, G_i\}.$$

The list of descendants in a block, denoted B , is defined as

$$B = \{D_0, D_1, D_2\}.$$

NOTE – $\{A, B\}$ denotes the concatenation of the lists A and B .

A shorthand notation for certain binary words that describe information about bit plane b for a list of coefficients Ψ is defined as follows:

- ✓ let $types_b[\Psi]$ denote the binary word consisting of the b^{th} magnitude bit of each coefficient x in Ψ such that $t_b(x)$ equals 0 or 1;
- ✓ let $signs_b(\Psi)$ denote the binary word consisting of the sign bit of each coefficient x in Ψ such that $t_b(x) = 1$, with a sign bit of '1' for negative coefficients and '0' for nonnegative coefficients;
- ✓ given a list of type values $\Lambda = \{\lambda_0, \lambda_1, \lambda_2, \dots, \lambda_l\}$, let $tword[\Lambda]$ denote the binary word consisting of the sequence of type values λ_i in Λ that are equal to 0 or 1.

The b^{th} magnitude bits for all AC coefficients that are type 0 at bit plane $b+1$ (i.e., not selected before the current bit plane) shall be communicated to the decoder by joining them to form binary words associated with each data type (parent, child, grandchild):

- ✓ $types_b[P]$;
- ✓ $types_b[C_i]$ for $i = 0, 1, 2$; and
- ✓ $types_b[H_{ij}]$ for $i = 0, 1, 2, j = 0, 1, 2, 3$.

At early bit planes, many sets of coefficients in a block tend to all be of type 0, and thus many of these words are initially all zeros. To effectively encode in this situation, the BPE shall make use of the following *transition* words to indicate when groups of coefficients at a lower depth are all Type 0:

- ✓ $tran_B = \text{null}$, if $tran_B = 1$ at any more significant bit plane,
 $tword[\{t_{\max}(B)\}]$, otherwise;
- ✓ $tran_D = tword[\{t_{\max}(D_i) : i=0,1,2, \text{ such that } t_{\max}(D_i) \neq 1 \text{ in all more significant bitplanes}\}]$;

- ✓ $tran_G = tword[\{tmax(G_i) : i=0, 1, 2, \text{ such that } tmax(D_i) > 0 \text{ in current or any more significant bit planes } \}]$;
- ✓ $tran_{Hi} = tword[\{tmax(H_{i0}), tmax(H_{i1}), tmax(H_{i2}), tmax(H_{i3})\}]$ for $i = 0, 1, 2$.

At bit plane b , the BPE shall use the following sequence of words, generated in three stages, to update all of the AC coefficients in the block that were Type 0 at the previous bit plane:

- **Stage 1 (parents):**

$types_b[P]$, $signs_b[P]$.

- **Stage 2 (children):**

1) $tran_B$;

2) $tran_D$, if $tran_B \neq 0$ and $t_{max}(B) \neq -1$;

3) $types_b[C_i]$ and $signs_b(C_i)$ for each i such that $t_{max}(D_i) \neq 0, -1$.

- **Stage 3 (grandchildren):**

If $tran_B = 0$ or $t_{max}(B) = -1$, then stage 3 is unnecessary and shall be omitted. Otherwise stage 3 consists of:

- ✓ $tran_G$;

- ✓ $tran_{Hi}$, for each i such that $t_{max}(G_i) \neq 0, -1$;

- ✓ $types_b[H_{ij}]$ and $signs_b[H_{ij}]$ for each i such that $t_{max}(G_i) \neq 0, -1$ and each j such that $t_{max}(H_{ij}) \neq 0, -1$.

Words $types_b[P]$, $types_b[C_i]$, $types_b[H_{ij}]$, $tran_D$, $tran_G$, $tran_{Hi}$ shall be entropy coded, i.e., each shall be replaced with a corresponding variable-length codeword, whenever such a word has a length of at least 2 bits.

5.3.3.3 Mapping Words to Symbols

The entropy coding procedure used to encode the words $types_b[P]$, $types_b[C_i]$, $types_b[H_{ij}]$, $tran_D$, $tran_G$, $tran_{Hi}$ shall be accomplished through the use of variable-length codes. Words having a length of one bit, and sign-bit words, shall be included in the compressed data stream without further coding. Words of length greater than one bit shall be coded in the sequence in which they occur within each stage with the entropy coding method.

The process of variable-length coding of these words shall follow a two-step process:

- first, the word values shall be mapped to integer values referred to as *symbols*; then
- each integer shall be encoded using a variable-length binary codeword.

Under the mapping, two-bit, three-bit, and four-bit words shall be mapped to symbols using table 5.1, 5.2, or 5.3, respectively.

Table5.1: Integer Mapping for Two-Bit Words

Word	Symbol
00	0
01	2
10	1
11	3

Table5.2 Integer Mapping for Three-Bit Words

Word	Symbol ($types_b[P]$, $types_b[C]$, $types_b[H]$, $tran_C$, $tran_H$)	Symbol ($tran_0$)
000	1	-
001	4	3
010	0	0
011	5	4
100	2	1
101	6	5
110	3	2
111	7	6

Table 5.3 Integer Mapping for Four-Bit Words

Word	Symbol ($types_b[C]$)	Symbol ($types_b[H]$, $tran_H$)
0000	10	-
0001	1	1
0010	3	3
0011	6	6
0100	2	2
0101	5	5
0110	9	9
0111	12	11
1000	0	0
1001	8	8
1010	7	7
1011	13	12
1100	4	4
1101	14	13
1110	11	10
1111	15	14

The mappings are intended to produce symbol values in order of decreasing frequency. (i.e., the most frequently occurring word is mapped to symbol value 0, the next most frequent to 1, etc.) This makes effective coding possible through the coding procedure because the codewords are arranged in order of increasing length.

5.3.3.4 Entropy Coding the Symbols

The symbols shall be encoded using the variable-length binary codes given in table 5.4, 5.5, and 5.6.

Within a gaggle,

- all two-bit words shall be encoded using one of the two variable-length code options given in table 5.4;
- all three-bit words shall be encoded using one of the three variable-length code options given in table 5.5;
- all four-bit words shall be encoded using one of the four variable-length code options given in table 5.6.

The variable-length codes used are permitted to change with each gaggle.

Table 5.4 : Variable Length Code Options for Two-Bit Words

Input Symbol	Code Option 0	Code Option 1 (uncoded)
0	1	00
1	01	01
2	001	10
3	000	11



P- 3463

Table 5.5 :Variable Length Code Options for Three-Bit Words

Input Symbol	Code Option 0	Code Option 1	Code Option 3 (uncoded)
0	1	10	000
1	01	11	001
2	001	010	010
3	00000	011	011
4	00001	0010	100
5	00010	0011	101
6	000110	0000	110
7	000111	0001	111

Table 5.6: Variable Length Code Options for Four-Bit Words

Input Symbol	Code Option 0	Code Option 1	Code Option 2	Code Option 3 (uncoded)
0	1	10	100	0000
1	01	11	101	0001
2	001	010	110	0010
3	0001	011	111	0011
4	000000	0010	0100	0100
5	000001	0011	0101	0101
6	000010	00000	0110	0110
7	000011	00001	0111	0111
8	00001000	000010	00100	1000
9	00001001	000011	00101	1001
10	00001010	000100	00110	1010
11	00001011	000101	00111	1011
12	00001100	0001100	00000	1100
13	00001101	0001101	00001	1101
14	00001110	0001110	00010	1110
15	00001111	0001111	00011	1111

For each word size within a gaggle (i.e., two-bit, three-bit, and four-bit words), the code option selected shall be the one that minimizes the encoded length of the gaggle when all of the words are coded. The code parameter optimizations are performed over all of the words in the gaggle even when the quality output mode specified ensures that some of those words are not included in the compressed bit stream. The code option selected shall be indicated using ID bits that take on the values given in table 5.7.

The uncoded option shall be selected whenever it minimizes the number of encoded bits, even if another option gives the same number of bits. When two or more code parameters minimize the number of encoded bits, the smallest code parameter option shall be selected.

Table 5.7: Identifying the Variable Length Code Options

No. of Bits for Mapped Patterns	No. of ID Bits	ID and Associated Code Option
2	1	0: code option 0 1: uncoded
3	2	00: code option 0 01: code option 1 11: uncoded
4	2	00: code option 0 01: code option 1 10: code option 2 11: uncoded

The ID bits for specifying coding options for words of a given length shall be inserted immediately preceding the first appearance of a codeword for a given length within a gaggle. When no word of the given length occurs in a gaggle, no ID bits shall be present.

5.3.4 Stage 4 Coding

In stage 4 of coding, the b^{th} magnitude bit of each AC coefficient x with type $t_b(x)=2$ shall be included in the output bit stream.

For each block, the output bit string shall consist of the b^{th} magnitude bit of type 2 coefficients, in the following order:

- p_i , for each $i = 0,1,2$;
- members of C_i , for each $i = 0,1,2$;
- members of H_{ij} , for each $i = 0,1,2$, and each $j = 0,1,2,3$.

The resulting strings for all blocks in the segment shall be concatenated to produce the entire stage 4 output string for the segment.

5.4 GENERATION OF TRANSITION WORDS OF A BLOCK BITPLANE-WISE – AN EXAMPLE

Computation of transition words of a block (8x8 coefficients) bitplane (BP)-wise by AC-BPE scan process is illustrated using an example shown below in Figure 5.13

13	-6	2	5	2	5	3	4
10	5	2	0	6	-5	4	3
3	-5	-1	3	6	4	2	3
0	0	3	0	-5	3	3	2
3	4	5	2	3	2	2	0
2	5	2	6	0	2	1	-3
4	3	-2	1	0	3	-2	0
1	4	0	0	-2	0	0	1

Figure 5.13 Block - 8x8 DWT Coefficients

The coefficients in each list are as follows:

Parent coefficients:

$$P = \{-6, 10, 5\}$$

Children coefficients:

$$C_0 = \{2, 5, 2, 0\}$$

$$C_1 = \{3, -5, 0, 0\}$$

$$C_2 = \{-1, 3, 3, 0\}$$

Grandchildren coefficients:

$$G_0 = \{2, 5, 6, -5, 3, 4, 4, 3, 6, 4, -5, 3, 2, 3, 3, 2\}$$

$$G_1 = \{3, 4, 2, 5, 5, 2, 2, 6, 4, 3, 1, 4, -2, 1, 0, 0\}$$

$$G_2 = \{3, 2, 0, 2, 2, 0, 1, -3, 0, 3, -2, 0, -2, 0, 0, 1\}$$

The maximum magnitude of AC coefficients is $p_1 = 10 (1010)_b$. Therefore, the AC coder scans from the fourth bit plane (MSB).

A. MSB Bitplane

Parent coefficients: $p_1 = 10 (1010)_b$ is significant at the bitplane, while $p_0 = -6 (0110)_b$ & $p_2 = 5 (0101)_b$ are insignificant. Therefore $\text{types}_b[P] = (010)_b$ & $\text{signs}_b[P] = 0$ as $p_1 > 0$.

Children & Grandchildren coefficients: As all the descendent coefficients are insignificant at MSB bitplane, $\text{tran}_B = 0$. The bit scan ends for MSB bit plane.

B. Third Bitplane

Parent coefficients: As p_1 has been marked significant at MSB bitplane, it is omitted at this bit plane at this stage. As $p_0 = -6$ and $p_2 = 5$ turn to significant at this bit plane, $\text{types}_b[P] = (11)_b$ & $\text{signs}_b[P] = (10)_b$ as $p_0 < 0$ and $p_2 > 0$.

Children coefficients: $\text{tran}_B = (1)_b$, as two children coefficients become significant. $\text{tran}_D = (110)_b$, as both C_0 and C_1 contain one coefficient that become significant while C_2 has no significant coefficients. The AC coder now scans the children coefficients in C_0 and C_1 . $\text{types}_b[C_0] = (0100)_b$ & $\text{signs}_b[C_0] = (0)_b$, as $C_0 = \{2, 5, 2, 0\}$. Similarly, $\text{types}_b[C_1] = (0100)_b$ & $\text{signs}_b[C_1] = (1)_b$, as $C_1 = \{3, -5, 0, 0\}$.

Grandchildren coefficients: $\text{tran}_G = (110)_b$, as two grandchildren coefficients become significant. $\text{tran}_{H0} = (1110)_b$, $\text{tran}_{H1} = (1110)_b$, as $H_{00}, H_{01}, H_{02}, H_{10}, H_{11}, H_{12}$ contain three coefficients that become significant while tran_{H2} has no significant coefficients. The AC coder now scans grandchildren coefficients in $H_{00}, H_{01}, H_{02}, H_{10}, H_{11}$ & H_{12} .

Table 5.8 -types_b and signs_b of Grandchildren coefficients 3rd BP

$\text{types}_b[H_{00}] = (0111)_b$	$\text{signs}_b[H_{00}] = (001)_b$
$\text{types}_b[H_{01}] = (0110)_b$	$\text{signs}_b[H_{01}] = (00)_b$
$\text{types}_b[H_{02}] = (1110)_b$	$\text{signs}_b[H_{02}] = (001)_b$
$\text{types}_b[H_{10}] = (0101)_b$	$\text{signs}_b[H_{10}] = (00)_b$
$\text{types}_b[H_{11}] = (1001)_b$	$\text{signs}_b[H_{11}] = (00)_b$
$\text{types}_b[H_{12}] = (1001)_b$	$\text{signs}_b[H_{12}] = (00)_b$

AC refinement: As P_1 has been marked significant at MSB bitplane, its bit at the third bitplane, $(0)_b$ is sent to bit stream without coding. Coding of the third bitplane ends.

C. Second Bitplane

Parent coefficients: As p_0 , p_1 and p_2 have been marked significant at the third and fourth bit planes, no coding is necessary at this stage.

Children coefficients:

Table 5.9 - Children coefficients 2nd BP

tran _B is omitted	
tran _D = $(1)_b$	
types _b [C ₀] = $(110)_b$	signs _b [C ₀] = $(00)_b$
types _b [C ₁] = $(100)_b$	signs _b [C ₁] = $(0)_b$
types _b [C ₂] = $(0110)_b$	signs _b [C ₂] = $(00)_b$

Grandchildren coefficients:

Table 5.10 - Grandchildren coefficients 2nd BP

tran _G = $(1)_b$,	
tran _{H0} = $(1)_b$,	
tran _{H1} = $(1)_b$,	
tran _{H2} = $(1111)_b$	
types _b [H ₀₀] = $(1)_b$	signs _b [H ₀₀] = $(0)_b$
types _b [H ₀₁] = $(11)_b$	signs _b [H ₀₁] = $(00)_b$
types _b [H ₀₂] = $(1)_b$	signs _b [H ₀₂] = $(0)_b$
types _b [H ₀₃] = $(1111)_b$	signs _b [H ₀₃] = $(0000)_b$
types _b [H ₁₀] = $(11)_b$	signs _b [H ₁₀] = $(00)_b$
types _b [H ₁₁] = $(11)_b$	signs _b [H ₁₁] = $(00)_b$
types _b [H ₁₂] = $(10)_b$	signs _b [H ₁₂] = $(0)_b$
types _b [H ₁₃] = $(1000)_b$	signs _b [H ₁₃] = $(1)_b$
types _b [H ₂₀] = $(1101)_b$	signs _b [H ₂₀] = $(000)_b$
types _b [H ₂₁] = $(1001)_b$	signs _b [H ₂₁] = $(01)_b$
types _b [H ₂₂] = $(0110)_b$	signs _b [H ₂₂] = $(01)_b$
types _b [H ₂₃] = $(1000)_b$	signs _b [H ₂₃] = $(1)_b$

AC refinement:

Table 5.11-AC refinement 2nd BP

Parent	(110) _b
Children	(00) _b
Grandchildren	(01000100000100) _b

D. LSB bit plane

All parent, children and grandchildren coefficients have been marked significant at MSB, third and second bitplanes, no coding is necessary at LSB bitplane.

AC refinement:

Table 5.12-AC refinement LSB BP

Parent	(001) _b
Children	(0101111) _b
Grandchildren	(010110010011011010011000010 010001100) _b

The transition words (2, 3, 4 bit words) generated will not be directly sent to bitstream. These words are first mapped to symbols and further entropy coded using variable length code option which produces lesser number of bits.

CHAPTER 6

AC BITPLANE ENCODER-SERIAL ARCHITECTURE

6.1 PROPOSED SERIAL ARCHITECTURE

The functional blocks of the proposed serial architecture is shown in Fig .6.1 below. AC coefficients are fed into gaggle memory block wise in ping-pong fashion i.e., the first gaggle memory stores 16 blocks and the next 16 blocks in the second gaggle memory as shown in the Fig. 6.1. Transition words are generated for every block in three stages to update all the AC coefficients in the block. Preprocessing stage computes the coefficient significance of the current bitplane which is derived from the higher magnitude bits of the AC coefficient. The bitplane enable provides the number of significant bitplanes information. The symbol mapper maps the transition words generated at each stage to the integer values referred to as symbols. Further, code option is found out iteratively which produces less compressed data.

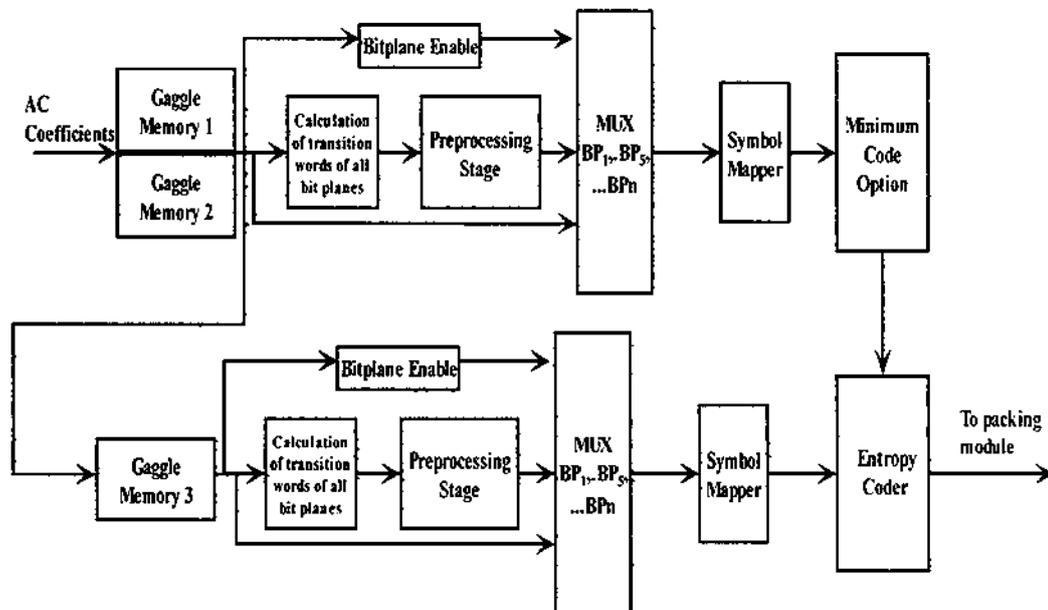


Fig.6.1- Proposed Serial Architecture

The entropy coded AC coefficients are packed to form a bit stream. The resulting bit stream constitutes an embedded data format that provides progressive transmission within a segment. Based on the minimum code option selected for each word size within a gaggle are entropy coded. Instead of storing the transition words input data itself stored in gaggle memory 3 by doing so memory requirement is largely decreased. Preprocessing stage, calculation of transition words, bitplane enables modules are repeated because these modules requires only few OR gates for implementation.

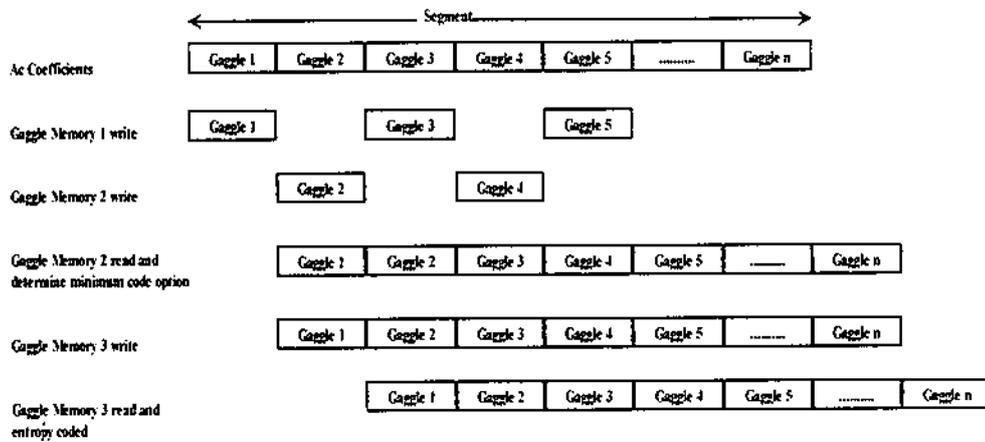


Fig.6.2 Timing Diagram

CHAPTER 7

RESULTS AND CONCLUSION

The aim of the project is to design hardware efficient AC Bitplane Encoder -Serial Architecture. Each block in the module implemented as described in Chapter 6 should perform properly to produce the bit stream. The function of each module can be validated from the functional simulation results. Sixty three AC coefficients are given as the input to the AC bitplane encoder. These coefficients are entropy coded and packed to form a bit stream. The resulting bit stream constitutes an embedded data format that provides progressive transmission within a segment. From the simulation results we can draw the conclusion for the implemented design.

7.1 SIMULATION RESULTS

The 63 AC coefficients from the DWT module is given as the input to the AC –BPE module. The simulation results of the different blocks realized in the design is discussed here.

7.1.1 COEFFICIENT COUNTER & BLOCK COUNTER & GAGGLE COUNTER

The coefficient counter is designed to count 64 coefficients which can be achieved by designing a 6 bit counter as shown in fig.7.1. A block consists of 64 coefficients and the block counter increments for every 64 coefficient count. A group of 16 blocks form a gaggle. For every 16 block count gaggle count is incremented. Source coding is done with respect to gaggle.

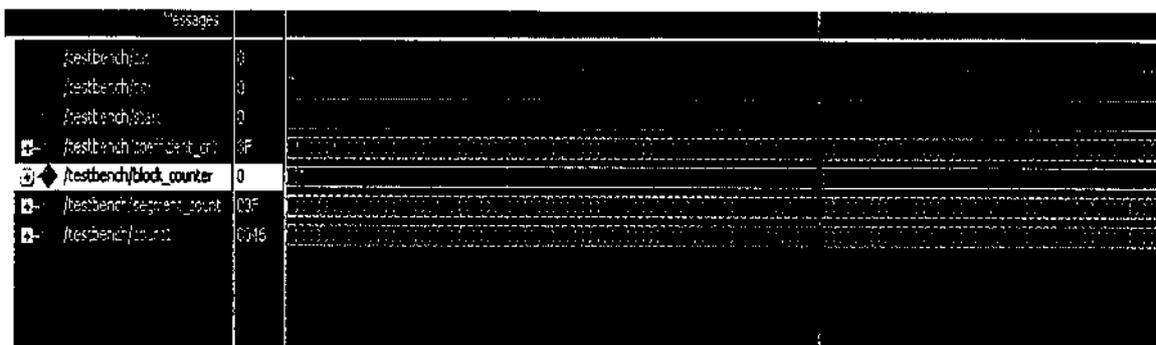


Figure 7.1 Coefficient counter waveforms

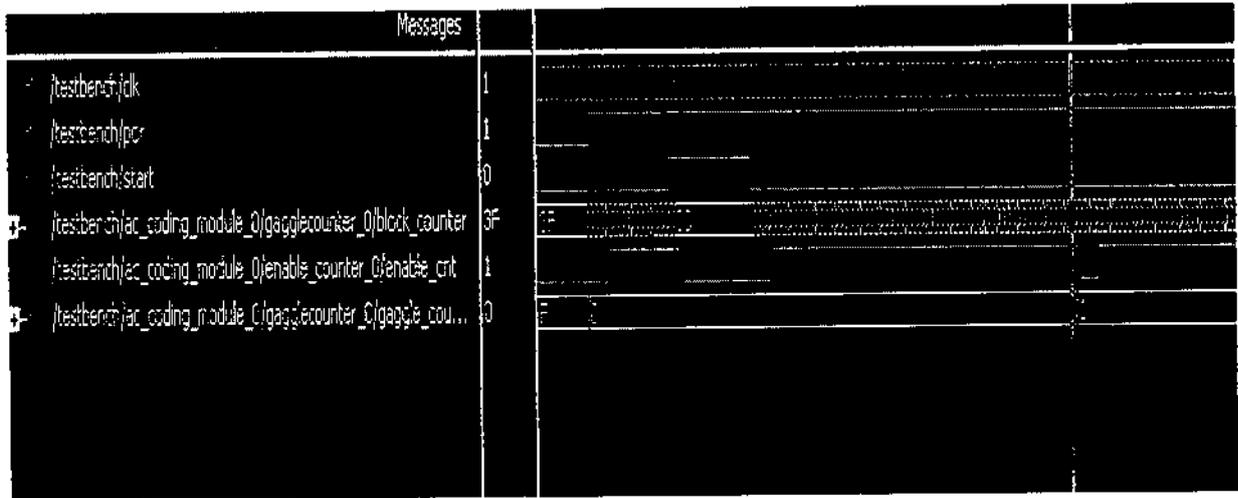


Figure 7.2 Block counter & Gaggle counter waveforms

7.1.2 ADDRESS GENERATION & GAGGLE MEMORY

The write and read addresses are generated with respect to clk1() and clk2() respectively. Reading is performed 16 times faster than writing. Two DPRAM's (Dual Port Random Access Memory) are designed in ping pong fashion i.e., first gaggle memory stores 16 blocks and the next 16 blocks in the second gaggle memory.

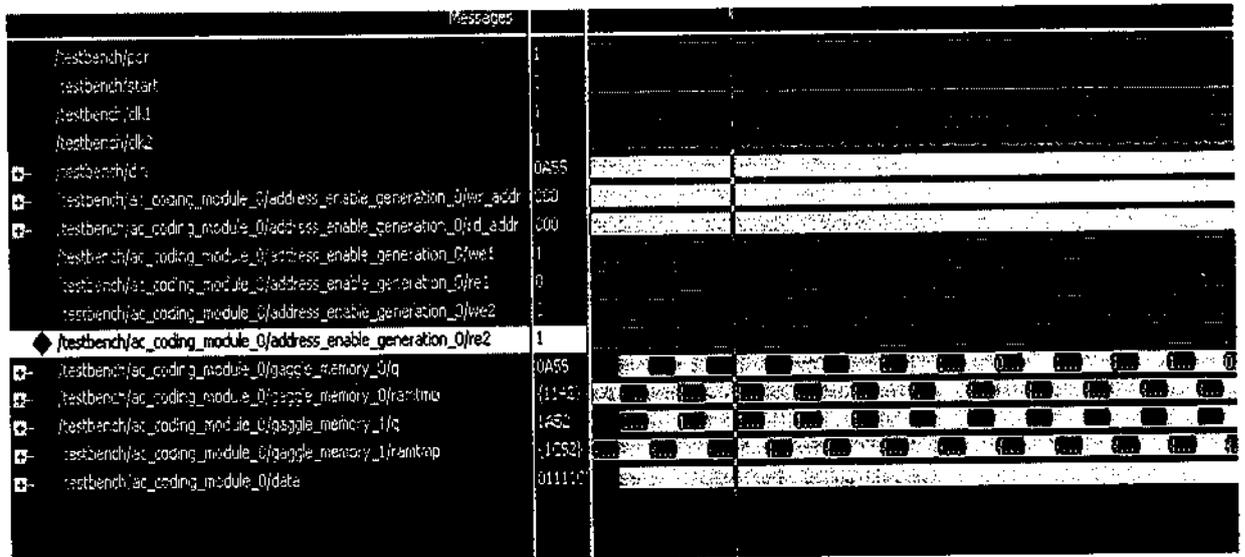


Figure 7.3 Address Generation & Gaggle Memory

7.1.3 INPUT DATA LATCH OF PARENT, CHILDREN & GRANDCHILDREN COEFFICIENTS

AC coefficients data is read from the input text file. The parent coefficients (i.e., p1,p2 & p3) and children and grandchildren coefficients(i.e.,cg1, cg2, cg3, cg4) are latched.



Figure 7.4 Input Data Latch waveforms

7.1.4 TRANSITION WORDS OF tran D , tran G & tran H

Transition words are generated for every block in three stages to update all the AC coefficients in the block .The transition words generated will not be directly sent to bitstream.

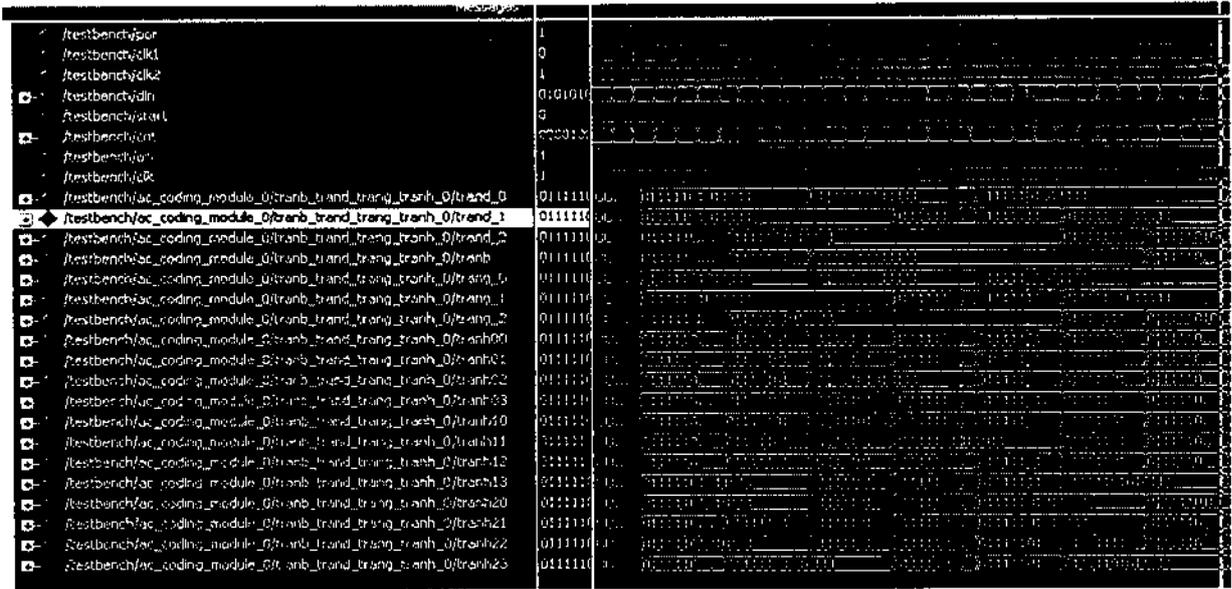


Figure 7.5 Transition words waveforms

7.1.5 PREPROCESSING OF tran D

Preprocessing stage computes the coefficient significance of the current bitplane which is derived from the high magnitude bits of the AC coefficients. Preprocessing is performed by using OR gates. Preprocessing is done for all the transition words i.e. tran B, tran D, tran G, tran H.



Figure 7.6 Preprocessing of tran D waveforms

7.1.6 SYMBOL MAPPER

The symbol mapper maps the transition words generated at each stage to the integer values referred as symbols

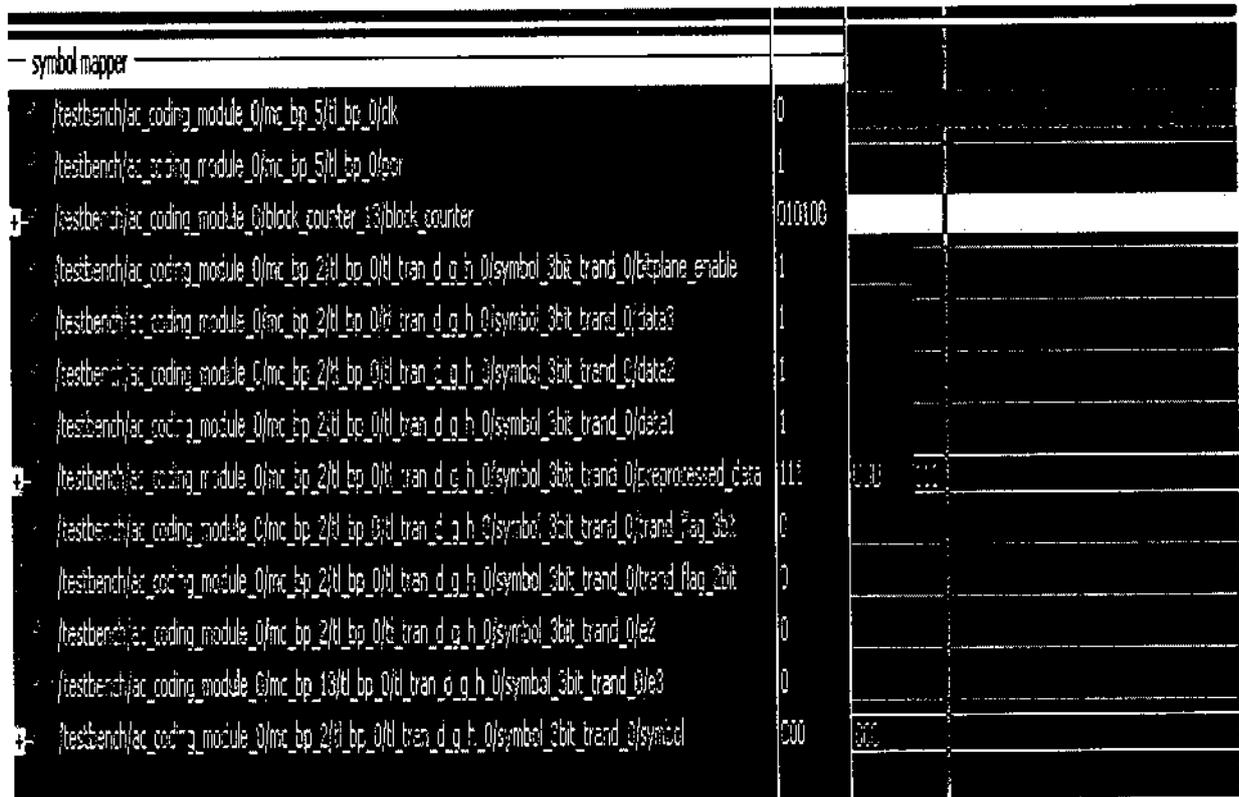


Figure 7.7 Symbol Mapper waveforms

7.1.7 MINIMUM CODE OPTION SELECTION

Minimum code option is done with respect to gaggle. Minimum code option is found out iteratively which produces less compressed data.

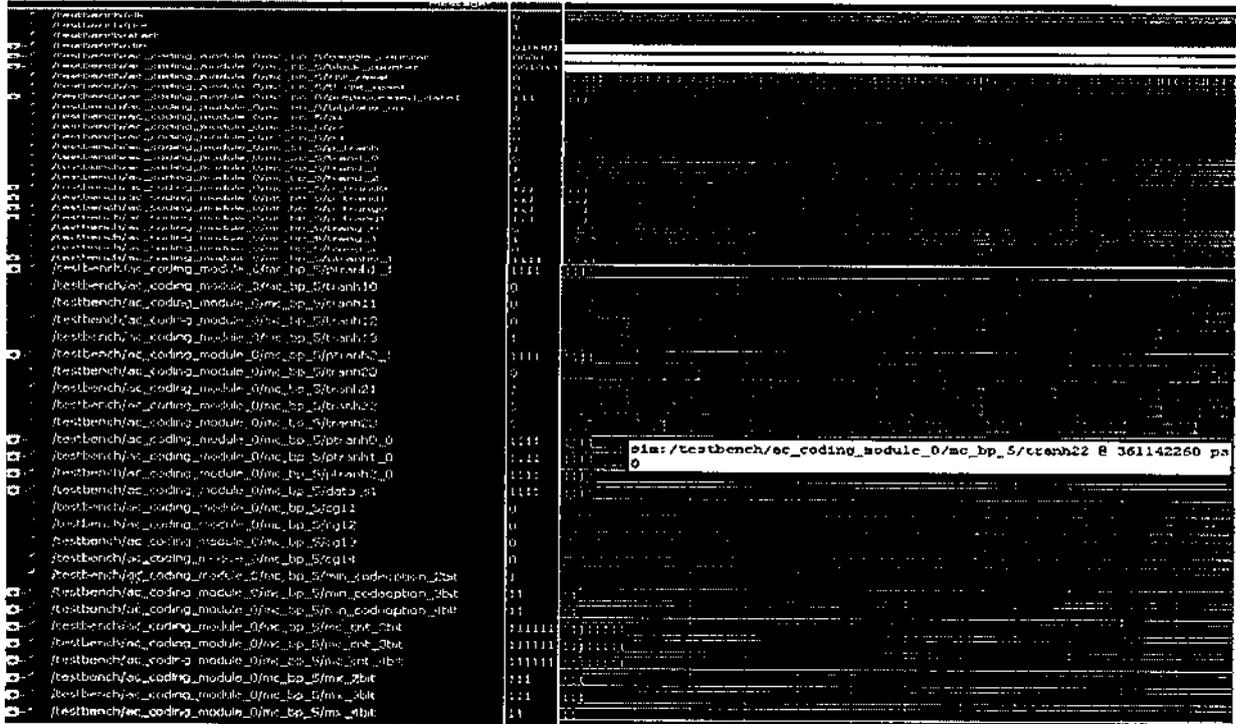


Figure 7.8 Minimum code option waveforms

7.2 SYNTHESIS REPORT

Serial Architecture of AC Bitplane Encoder is synthesized and the following are the outcomes and the snapshot of synthesis report is shown in the following figures. The synthesis report gives the utilization of the device for the module design. The total logic gates and the RAM used for the design are also listed in the report. The following are the highlights of the designed modules obtained from the synthesis report.

GAGGLE MEMORY

Timing Report

Synplify Pro D-2009.12A [D:/serial_ac_coding/CCSDS_BPF/synthesis/GAGGLE_MEMORY.htm]

File Edit View Project Run Analysis HDL-Analyst Options Window Tech-Support Web Help

Performance Summary

Worst slack in design: 7.661

Starting Clock	Requested Frequency	Estimated Frequency	Requested Period	Estimated Period	Slack	Clock Type	Clock Group
GAGGLE_MEMORY clk2	100.0 MHz	427.5 MHz	10.000	2.339	7.661	inferred	Inferred_clkgroup_1

Device Utilization

Target Device	rtax4000s_cqfp352-1
Combinational Cells	10459 of 40320 (26%)
Sequential Cells	14352 of 20160 (71%)
Total Cells	24811 of 60480 (42%)
DSP Blocks	0
Clock Buffers	2
IO Cells	52

Transition Words

Timing Report

Performance Summary

Worst slack in design: 7.601

Starting Clock	Requested Frequency	Estimated Frequency	Requested Period	Estimated Period	Slack	Clock Type	Clock Group
tranB_tranD_tranG_tranHclk	100.0 MHz	416.8 MHz	10.000	2.399	7.601	inferred	Inferred_clkgroup_0

Device Utilization

Target Device	rtax4000s_cqfp352-1
Combinational Cells	574 of 40320 (1%)
Sequential Cells	560 of 20160 (3%)
Total Cells	1134 of 60480 (2%)
DSP Blocks	0
Clock Buffers	1
IO Cells	365

Preprocessing Stage

Device Utilization

Target Device	rtax4000s_cqfp352-1
Combinational Cells	60 of 40320 (0%)
Sequential Cells	42 of 20160 (0%)
Total Cells	102 of 60480 (1%)
DSP Blocks	0
Clock Buffers	1
IO Cells	85

Minimum Code Option

Timing Report

The screenshot shows the Synplify Pro D-2009.12A software interface. The title bar reads "Synplify Pro D-2009.12A - [D:/serial_ac_coding/CCSDS_BPE/synthesis/MC_BP.htm]". The menu bar includes "File", "Edit", "View", "Project", "Run", "Analysis", "HDL-Analyst", "Options", "Window", "Tech-Support", "Web", and "Help". A toolbar with various icons is visible below the menu bar. The main window displays a "Performance Summary" window with the following content:

```
Performance Summary
*****

Worst slack in design: 3.610

Starting Clock      Requested  Estimated  Requested  Estimated  Slack  Clock  Clock
Frequency           Frequency  Frequency  Period     Period     Slack  Type   Group
-----
MC_BPclk            100.0 MHz  156.5 MHz  10.000     6.390     3.610  inferred  Inferred_clkgroup_0
```

Device Utilization

Target Device	rtax4000s_cqfp352-1
Combinational Cells	1016 of 40320 (3%)
Sequential Cells	675 of 20160 (3%)
Total Cells	1691 of 60480 (3%)
DSP Blocks	0
Clock Buffers	4
IO Cells	124

CHAPTER 8

CONCLUSION & FUTURE SCOPE

8.1 CONCLUSION

This thesis has presented a method for spacecraft on-board compression. The data compression algorithm recommended by CCSDS has benefited and will benefit many space missions by either reducing bandwidth, on-board storage requirement, or by increasing science data return. The percentage data reduction may not be the best achievable considering all other available techniques. However, its simplicity and adaptivity does allow high-speed spacecraft on-board compression of image data.

The algorithm reduces on-board implementation complexity. The BPE produces an encoded bitstream providing progressive transmission within a coded segment. This technique produces an embedded bitstream with the desirable features of precise rate control and no operator intervention. The algorithm is applicable to a variety of imaging instruments, and is suitable for push-broom sensors requiring immediate processing of data.

CCSDS based bitplane encoder is simulated in VHDL. The algorithm is implemented on hardware. The proposed serial architecture for AC bitplane encoder of CCSDS compression standard is the best choice for fast and reduces hardware requirements. The architecture also largely decreases the memory and logic circuit requirement for low data rate space applications.

8.2 FUTURE SCOPE

The extensions to the work completed so far are proposed in this section.

1. Place and route of the design has to be done.
2. A feature to compress color images has to be incorporated in the algorithm.

REFERENCES

- [1] Rafael C. Gonzalez, Richard E. Woods, "*Digital Image Processing*", Second Edition, 2002, Prentice-Hall, Inc.
- [2] "*Image Data Compression – Recommendation for Space Data System Standards*, CCSDS 122.0-B-1. Blue Book –Issue 1, Washington, D.C: CCSDS, November 2005.
- [3] Pen-Shu Yeh, Philippe Armbruster, Aaron Kiely, Bart Masschelein, Gilles Moury, Christoph Schafer, "*The New CCSDS Image Compression Recommendation* ", IEEE Aerospace Conference, Big Sky, Montana, Mark 5-12, 2005.
- [4] Tinku Acharya, Ping-Sing Tsai, "*JPEG2000 Standard for Image Compression: Concepts, Algorithms and VLSI Architectures*", Wiley- Interscience, John Wiley & Sons, Inc., Publication, 2005.
- [5] "*Image Data Compression – Report Concerning Space Data System Standards*, CCSDS 120.1-G-1. Green Book Book" –Issue 1, Washington, D.C: CCSDS, June 2007.
- [6] S. Jayaraman, S. Esakkirajan, T. Veerakumar, "*Digital Image Processing*", 2009, Tata McGraw Hill.70
- [7] Hongqiang Wang, "*Robust Image and Video Coding With Adaptive Rate Control*", IEEE Transaction 2009.
- [8] "CCSDS Documents", <http://www.ccsds.org/>
- [9] "Technical Documentation", <http://www.actel.com/>
- [10] <http://ntrs.nasa.gov/>

APPENDIX A

DEFINITIONS

In this document, for any real number x , the smallest integer n such that $n \geq x$ by

$$n = \text{ceil}(x)$$

SYMBOLS USED IN CODING STAGES

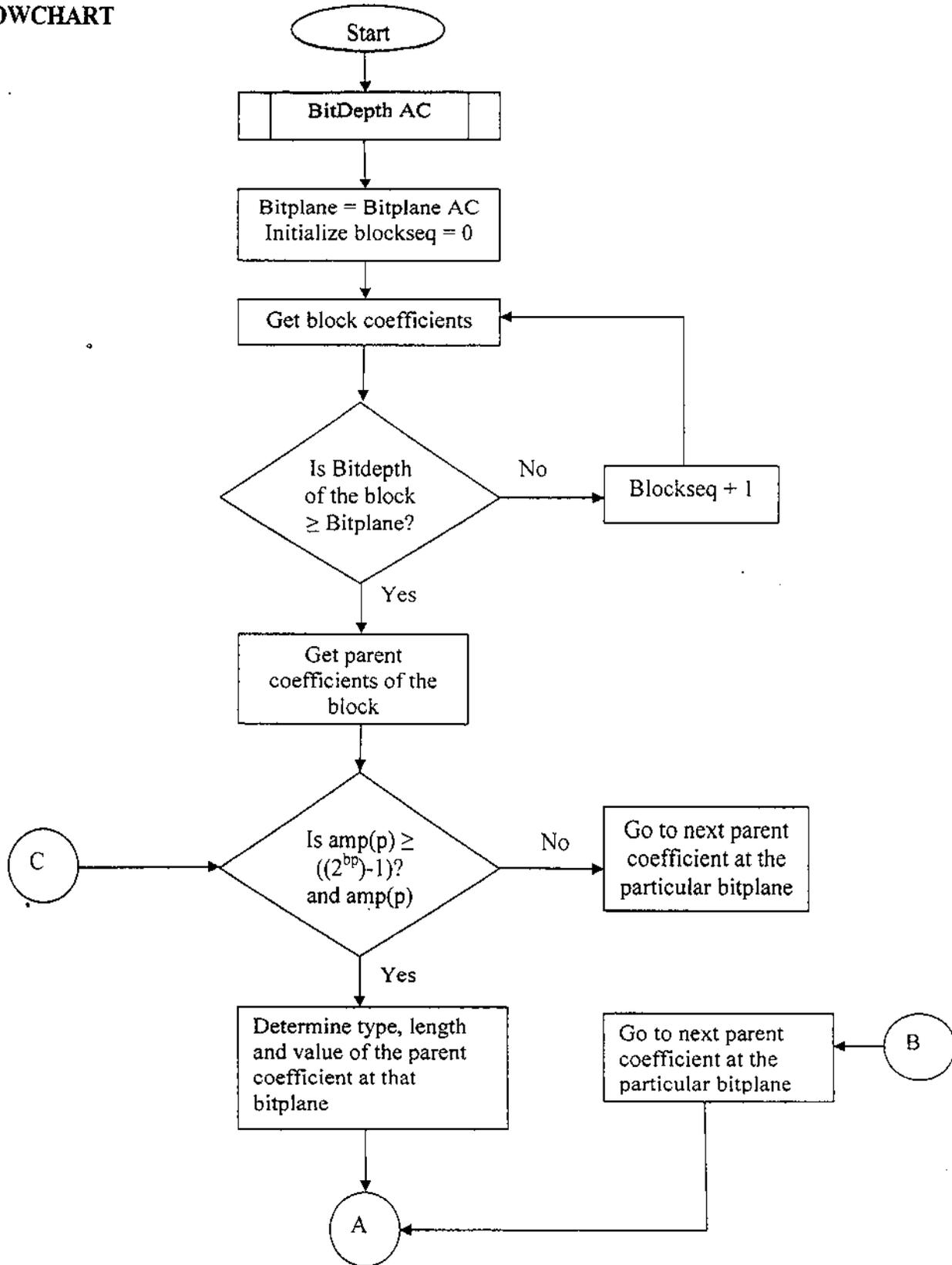
Table A-1: Symbols Used in Coding Stages

Category	Symbols	Meaning
wavelet coefficients	DC, c	DC coefficient
	c'	quantized DC coefficient
	AC, x	AC coefficient
	p	parent coefficient ($i=0, 1, 2$)
	L	DWT subband
ordered sets of wavelet coefficients assigned to a block	Ψ	variable set
	B	set of all descendent coefficients
	P	set of parent coefficients
	D_i	set of descendent coefficients ($i=0, 1, 2$)
	C_i	set of children coefficients ($i=0, 1, 2$)
	G_i	set of grandchildren coefficients ($i=0, 1, 2$)
	H_i	subset of set of grandchildren G_i ($i=0, 1, 2$; $j=0, 1, 2, 3$)
integer valued functions	$t_b(x)$	value indicating type of coefficient x , with respect to current bit plane with index b
	$t_{\max}(\Psi)$	$\max\{t_b(x) : x \in \Psi\}$
	$types_b[\Psi]$	binary word whose bits are the types $t_b(x)$ of the coefficients $x \in \Psi$, ignoring coefficients with types other than $t_b(x)=0$ or $t_b(x)=1$
	$signs_b[\Psi]$	binary word whose bits are the signs of the coefficients $x \in \Psi$, ignoring coefficients whose types are not $t_b(x)=1$. Positive signs give 0, negative signs give 1.
	$tword\{t_b, t_1, \dots, t_i\}$	binary words whose bits are the type values $\{t_b, t_1, \dots, t_i\}$, ignoring types whose values are not 0 or 1
binary words	$tran_B$	transition word for set B
	$tran_D$	transition word for collection of sets D_0, D_1, D_2
	$tran_G$	transition word for collection of sets G_0, G_1, G_2
	$tran_H$	transition word for collection of sets H_0, H_1, H_2, H_3 ($i=0, 1, 2$)

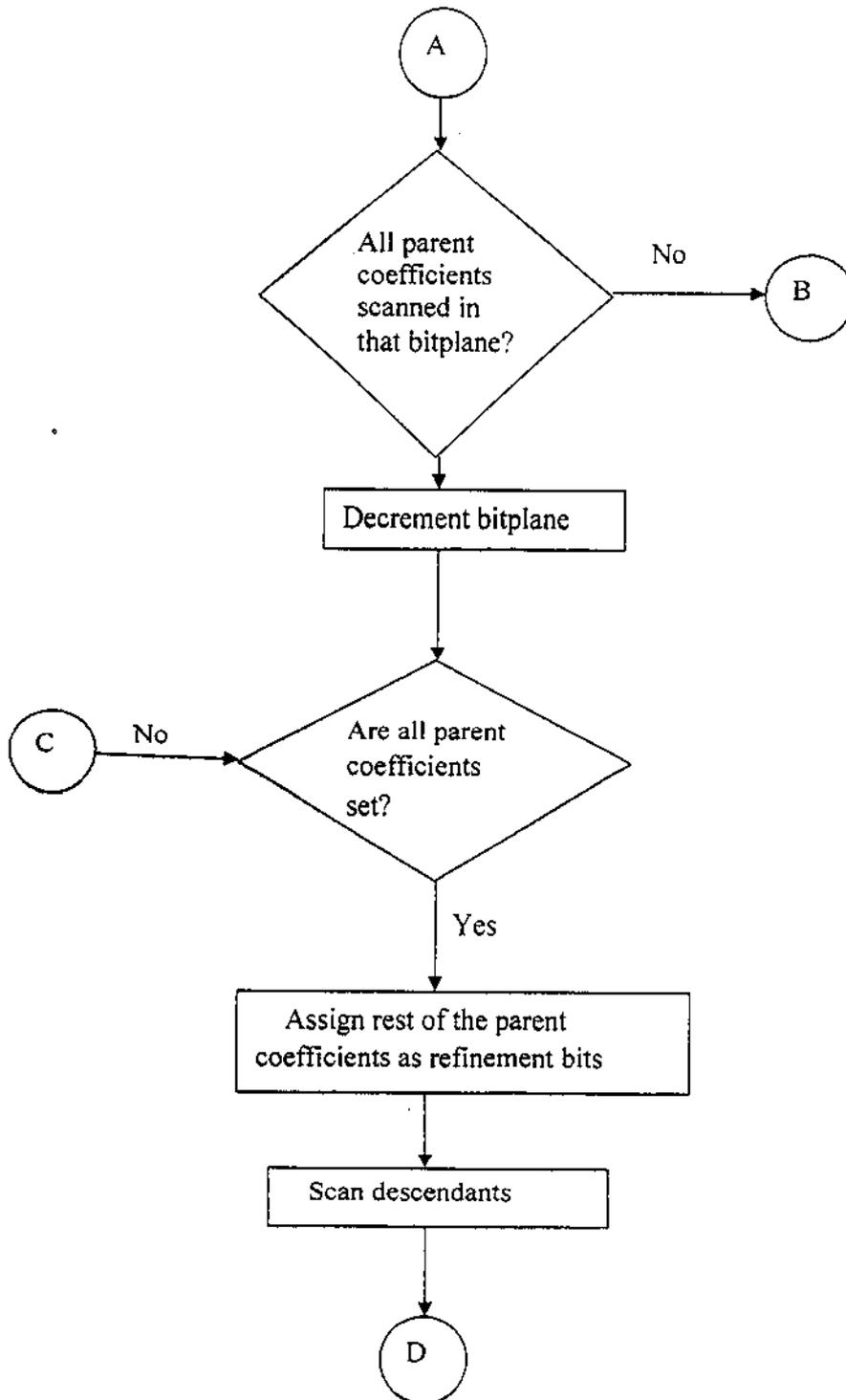
GLOSSARY OF ACRONYMS AND TERMS

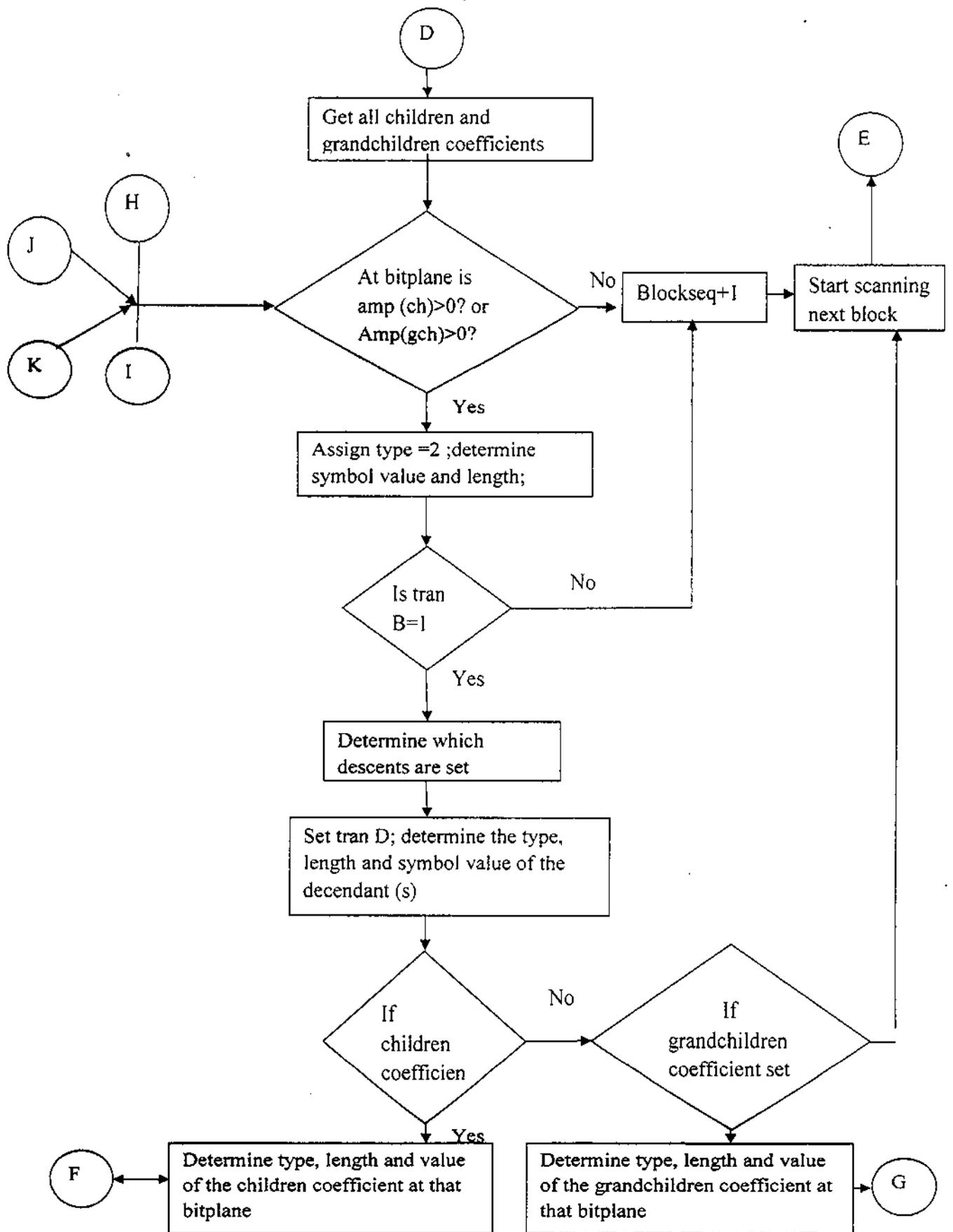
BPE	Bit-plane encoder: recommended processing algorithm used to compress wavelet coefficient data.
DWT	Discrete Wavelet Transform: recommended processing algorithm to transform image data to wavelet coefficient data.
MSB	Most Significant Bit (of a word): left-most bit in figures: highest power of two in binary representation; first bit to be transmitted in serial output.
DC coefficient	any wavelet coefficient from the lowest frequency subband LL3.
AC coefficient	any wavelet coefficient from any subband except LL3.
block	Collection of 64 wavelet coefficients, consisting of one DC coefficient from the LL3 subband and 63 uniquely associated AC coefficients from the remaining nine subbands. For encoding, the complete wavelet domain is divided into blocks that are pairwise disjoint. There is one block for each DC coefficient.
segment	The data field contains the encoded bits from S consecutive blocks. S is a userselected parameter such that $16 \leq S \leq 2^{20}$.
Gaggle	A gaggle consists of a set of consecutive blocks within a segment. Specifically, a segment is partitioned into gaggles, with each gaggle consisting of 16 blocks, except possibly the last gaggle, which contains $(S \bmod 16)$ blocks when S is not a multiple of 16.

FLOWCHART

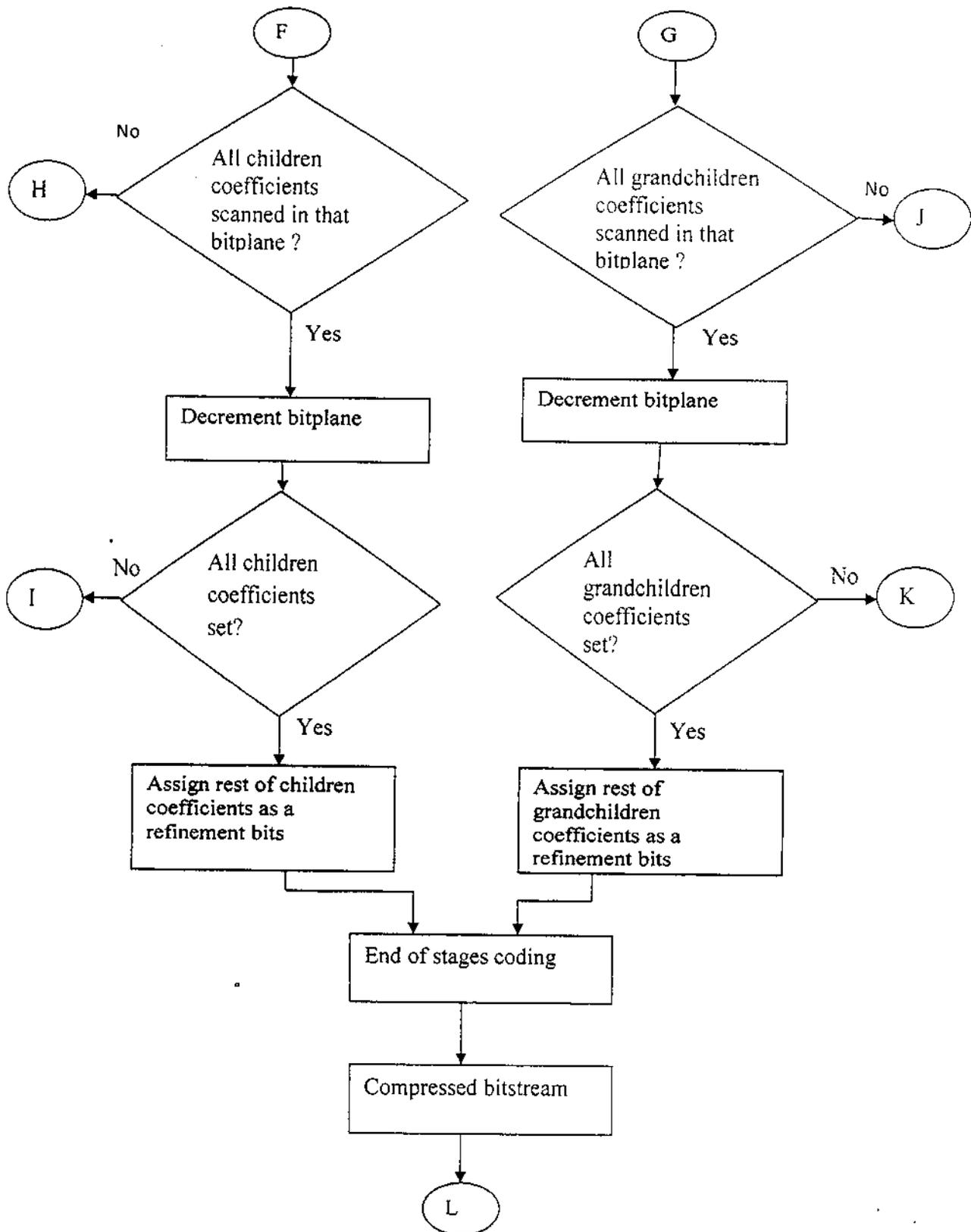


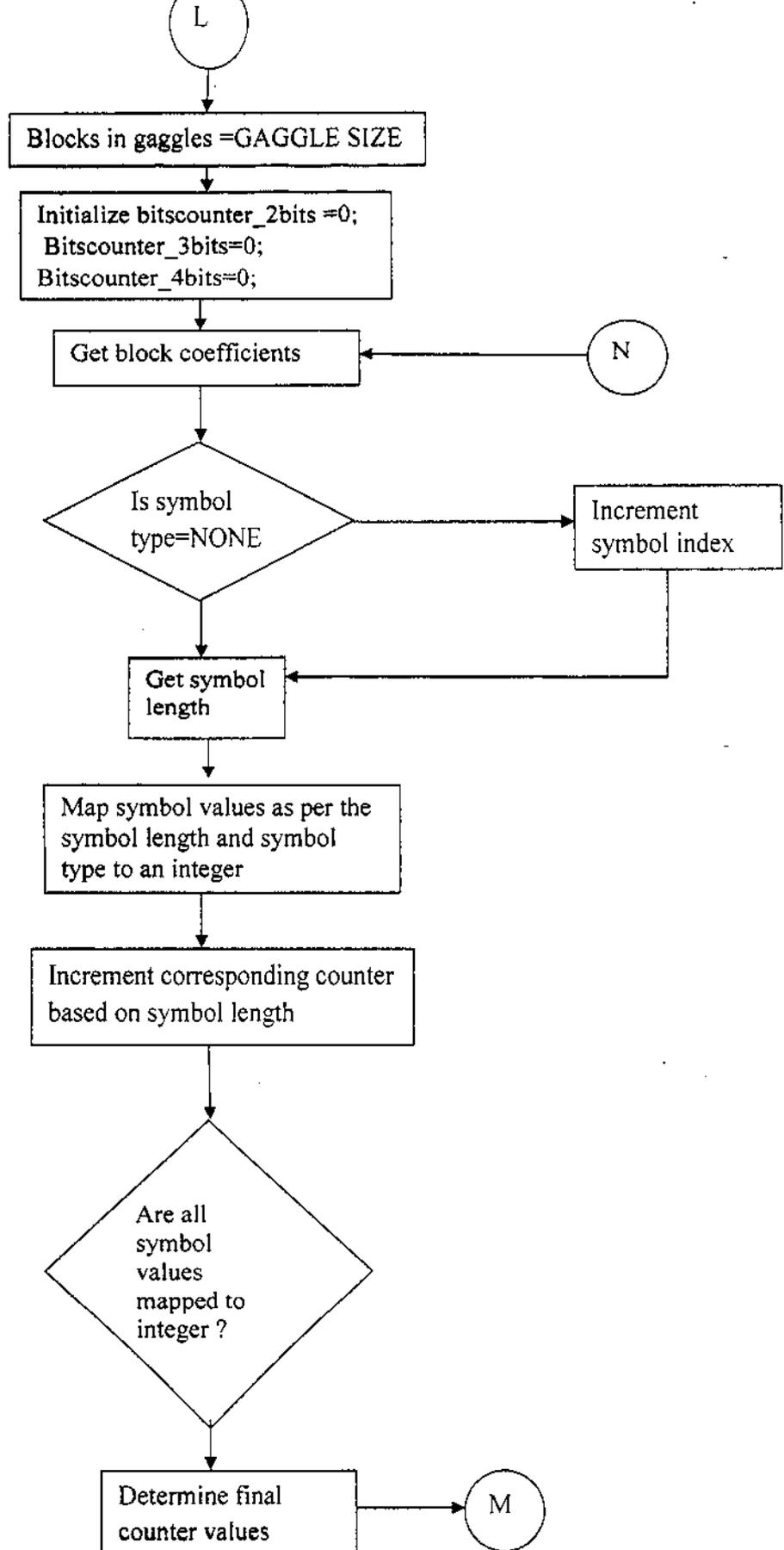
Flow chart for stage 1 coding of BPE





Flowchart for stage 2 and stage 3 coding of BPE





Flowchart for stage 4 Coding

