N93-25928

THE IMPLEMENTATION OF A LOSSLESS DATA COMPRESSION MODULE IN AN ADVANCED ORBITING SYSTEM: ANALYSIS AND DEVELOPMENT

Pen-Shu Yeh†, Warner H. Miller†, Jack Venbrux‡, Norley Liu‡, Robert F. Rice*

†Goddard Space Flight Center, Greenbelt, MD 20771

‡Microelectronics Research Center[1], University of Idaho, Moscow, ID 83843

*Jet Propulsion Laboratory, Pasadena, CA 91109

Abstract

Data compression has been proposed for several flight missions as a means of either reducing onboard mass data storage, increasing science data return through a bandwidth constrained channel, reducing TDRSS access time, or easing ground archival mass storage requirement. Several issues arise with the implementation of this technology. These include the requirement of a clean channel, onboard smoothing buffer, onboard processing hardware and on the algorithm itself, the adaptability to scene changes and maybe even versatility to the various mission types.

This paper gives an overview of an ongoing effort being performed at Goddard Space Flight Center for implementing a lossless data compression scheme for space flight. We will provide analysis results on several data systems issues, the performance of the selected lossless compression scheme, the status of the hardware processor and current development plan.

1 Introduction

Before implementing a data compression scheme onboard a spacecraft, it is important to address issues in the telecommunication channel and the architecture of the data system in which it resides. The advanced orbiting systems of the 1990's and beyond will demand a communication network which can support a wide range of data rates, complex international constellations of space platforms, extensive onboard computer networking and possibly cross-support among missions. To meet the requirement of such a network and data system, the Consultative Committee for Space Data Systems (CCSDS) has published a recommended standard: "Advanced Orbiting Systems, Networks and Data Links: Architectural Specification" [1], to provide descriptions of the architecture of a network and data structure recommended for future orbiting platforms.

An important feature of such a data system is that all sensor data are packetized into the hierarchical structure shown in Fig. 1. Sensor data are first encapsulated into a CCSDS packet of length up to 2^{16} bytes. It is further multiplexed with other CCSDS packets originated from other paths into Multiplexing Protocol Data Unit (MPDU) of fixed length. After being padded with error protection bits and other inserted data, the MPDU is further assigned a Virtual Channel, and converted into a Virtual Channel Data Unit (VCDU). Again, this is a fixed length data unit.

[1]Supported in part by NASA under grant NAGW-1406

The subsequent stage which takes MPDUs from both single and multiple sources is the Virtual Channel Access (VCA). Its output data rate is the fixed link data rate assigned to the platform. The VCA may accept inputs from various data sources of variable data rates and bandwidths. To provide a constant output data rate from VCA, a smoothing data buffer is required and occasionally fill VCDUs are transmitted when the buffer experiences data underflow.

An efficient lossless data compression scheme will almost always produce variable length coded bits. These coded bits will be concatenated to form CCSDS packets, which consequently will be of variable lengths too. This requires that the multiplexing unit be provided with smoothing buffer in order that the output MPDU rate is constant. Associated with the selected buffer size, there will also be instances when a fill MPDU is necessary. A filled data unit can be regarded as a decrease in the channel utilization or efficiency.

In the sequel, we will first address data system issues related to implementing onboard data compression. A description and performance analysis of a selected lossless compression algorithm will be given. It will be followed by the results of an ASIC development of this algorithm. Finally, a brief summary of our current efforts will be given.

2 Systems Issues

2.1 Clean Channel Requirement

As pointed out in [2] of the last Data Compression Workshop held at Snowbird, April 11, 1991, one characteristic of compressed data is its sensitivity to noise. That is, one bit in error can result in a burst of data errors during the reconstruction process. This sensitivity to noise results from the fact that most compression algorithms reconstruct data based on values from more than one sample. Specifically, it is apparent that for algorithms which perform Differential Pulse Code Modulation (DPCM) as a front end process, a reconstruction error will tend to propagate to the end of a packet. In general, a tradeoff exists between choosing a suitable packet length to match the telecommunication channel characteristics and the ease of interfacing instruments within a data system. However, the channel coding recommended by CCSDS employing a concatenated error control coding scheme of an outer Reed-Solomon (255, 223) code and a rate 1/2 convolutional inner code [3], will provide a channel with bit error rate (BER) much lower than 10^{-9} at SNR of 3 db. Operational use of this concatenated system should typically yield even lower error rate, far lower than the stated requirement of 10^{-6} for the compressed data [2].

2.2 Buffer Location, Requirement and Channel Efficiency

Lossless data compression methods, by which redundancy is removed from the source data, result in variable length bit strings which can be packetized. The variable length CCSDS packets are first enclosed in fixed length MPDUs. These MPDUs are input to the VCA either synchronously or asynchronously as shown in Fig. 2. For the synchronous sampling by the VCA: a MPDU packet consisting of either valid CCSDS packets or fill bits is passed to the VCA, at every sampling period t_s. In this scheme, the smoothing buffer is provided at the MPDU generator location. For the asynchronous sampling scheme: a MPDU is provided to the VCA only when it is filled with valid CCSDS packets. Therefore, the input to the VCA is sampled at variable time intervals which are a multiple t_p, the CCSDS packet generation period. The constant downlink data rate is achieved by providing a buffer at the VCA. The system sampling time t_s is determined by the data rate allocated to a specific instrument, while the packet generation period t_p relates to a sensor's data collection scheme.

During data underflow, filled MPDUs for Fig. 2(a), or filled CVCDUs for Fig. 2(b), are generated to maintain constant link rate. This causes reduced channel utilization.

For the smoothing buffer, its size requirement depends on the packet statistics. These effects have been simulated in a study [4] which shows that the long-term performance of both sampling strategies in Fig. 2 are similar. That is, the maximum buffer requirement and the channel efficiency are comparable. An example is given in Fig. 3, which shows these effects as a function of the sampling ratio t_s/t_p. The result was obtained by assuming a packet source of Gaussian distribution of mean packet length 1 MPDU and a σ of 0.1, which is related to the variation in source statistics. The performance is characterized in terms of the buffer length requirement (in MPDUs) and average fill fraction.

2.3 Recoverability

As mentioned earlier, a channel error on the compressed data bits is likely to cause reconstruction error that will also affect subsequent reconstructed data. This type of error propagation can be limited to the error within a single packet by employing a data compression scheme that resets at the beginning of every packet.

3 An Adaptive Lossless Source Coding Algorithm

In selecting a lossless compression algorithm for onboard applications, several criteria are considered:

Adaptivity: Spacecraft sensor data are usually characterized by wide variation in the statistics. Representative data come from Earth observation data over clouds, ocean, land, or spectrograph data of solar activities, or star fields, or galaxies. A selected algorithm should compress data at near optimal rate when the scene changes (even for one instrument) to fully exploit the benefit of data compression.

Ease of Implementation: For onboard implementation, an algorithm should require few processing steps, small memory, and insignificant power.

The universal source coding scheme, devised by Rice [5] [6] [7] [8] [9] was selected. Its function and performance are described in the following.

3.1 The Universal Source Coding Scheme

The Rice algorithm is a structure that provides efficient performance over a broad range of source entropy. This universality is accomplished by adaptively selecting the best of several options of an easily implemented variable length coding algorithm on the basis of a block of input samples. The size of the block is a compromise between algorithm adaptivity and the necessary overhead bits to identity algorithm options. Our earlier study has shown that a block size of 16 samples is optimal for most our test imagery. This block of input samples is pre-processed by first performing DPCM (or higher order prediction) and a mapping of the data into non-negative integers. A block diagram of the algorithm structure is provided in Fig. 4. One option of the algorithm codes these integers with a comma code, the other options are obtained by splitting a specified number of the least significant bits, k, off the integers, to be appended later to the comma code of the remaining most significant bits. These options are considered as coders running in parallel. The one that produces the least number of coding bits is selected and ID bits are generated to signal this option to the decoder.

3.2 Optimality of The Compression Algorithm

In an earlier analysis [10], it was shown that for source symbol sets having a Laplacian distribution, the first option is equivalent to a class of Huffman code under the Humblet condition. The other split-sample options are shown to be equivalent to the Huffman codes of a slightly modified Laplacian symbol set, at integer symbol entropy levels. For NASA's applications, especially on imagery, for which the symbol probabilities after DPCM are well modelled as Laplacian, the practical result is simple and profound: the Huffman code to use at each integer entropy value $(k+2)$ is the corresponding k split-sample option.

The theoretical performance of these split-sample options on a Laplacian symbol set is given in Fig. 5. As more split-sample options are included in the coding structure, the performance curve will be extended in the upper-right direction following the same trend.

A major advantage of this coding structure is that the codeword for each symbol is completely specified by knowing its order in the integer symbol set. No codebooks are needed, this significantly simplifies onboard hardware implementation.

3.3 Simulation and Comparison with Other Techniques

A set of nine test imagery of 128x128 pixels, acquired from NASA image library and shown in Fig. 7, was compressed using the selected algorithm. The top two rows are 8-bit data while the bottom row has 12-bit AVIRIS test data. The results are given in Fig. 6. The efficacy of the algorithm is clearly demonstrated.

In order to compare with other techniques, four other samples of University of Southern California (USC) 8-bit test images, shown in Fig. 8 are used. The results are listed in Tables 1, 2 and 3 in terms of three performance parameters: percentage reduction, the compression ratio (CR), and total coding bytes. For the Ziv-Lampel (LZ) algorithm, the *compress* utility on UNIX system is used [11]. The *pack* utility simulates the adaptive Huffman (AH in the Tables) code. The arithmetic coding (denoted as ARi in the Tables) scheme is implemented using [12]. To provide a fair basis for comparison, we also include results obtained by using these techniques on the same pre-processed, *i.e.* DPCM+mapping, imagery. It is expected that this pre-processing will largely de-correlate data and increase the performance of the three other techniques with which we are comparing the Rice algorithm. For the LZ, AH and ARi techniques, the results are listed under columns marked as p+LZ, p+AH and p+ARi.

It should be noted that the results for the Rice algorithm include an 8-bit reference for every scanline and a 3-bit ID for every 16 samples.

4 ASIC Development

An Application Specific Integrated Circuit (ASIC) chip set has been designed, fabricated and tested to perform the selected universal lossless compression algorithm [13] [14]. The general architecture follows Fig. 4.

4.1 General Descriptions of the Chip Set

The algorithm lends itself to a high speed integrated circuit implementation because:

1. The encoding process allows a highly pipelined architecture, and most of the decoding process can be pipelined as well.
2. Hardware can be shared inside the chips because the options are similar in structure.
3. No external RAM is needed to store tables or statistics.
4. No lookup tables are required on either the encoder or the decoder. The total on-chip RAM is only 320 bytes.

To allow easy interface with an onboard data system such as depicted in Fig. 2, the coded bits are preceded by a header word containing the total number of coding bits. It will be stripped by the packetizer before being concatenated with other blocks into CCSDS packets.

The default DPCM uses the previous value as a predictor, however, the design also allows an external reference to be used as predictor. The pre-processor functional module can also be by-passed completely to allow using only the entropy coding module. Because the encoder is designed to be able to operate continuously at the sample frequency, no sample buffer is needed to store scanlines. Features are also built in the decoder to prevent any decoding error induced by the channel noise to propagate to the next packet.

In order to adapt to a variety of potential instruments, the current chip set is designed to handle 4-14 bits of digital data. A 4-bit ID is attached to every block of 16 coded samples. In addition, reference samples can be inserted at a user-specified interval.

4.2 Chip Set Parameters

The chip set has been designed in a 1 micron CMOS process for low power consumption and high data rate. The resulting die area for both encoder and decoder was only 5mm on a side. Fig. 9 shows the chip plots for the encoder and decoder.

The designed chip set was fabricated and tested on a Hewlett Packard HP82000 IC tester with parametric tests and functional tests that use over 100,000 vectors. Table 4 lists its parameters.

4.3 Flight Readiness

The chip set, named Universal Source Encoder (USE) and Universal Source Decoder (USD), was fabricated using the Hewlett Packard commercial process line which was tested to withstand a total radiation dose of up to 1 Mrad. The USE chip will undergo thermal cycles and a vibration test as part of chip qualification before possible launch. Meanwhile, a rad-hard version of the USE/USD chip set will be developed before being installed in the flight data system.

5 Current Development Plan

Currently, a testbed for the USE chip is being designed. This testbed includes packetizer, multiplexer and interface to VCA on the encoder side. Plans have been made to perform end-to-end test through NASA's TDRSS and the NASCOM system, where the USD chip will be located to decode compressed data.

Acknowledgement

The authors would like to acknowledge the following personnel at Goddard Space Flight Center for participating in the design of the testbed: Wai Fong, Joe Miko, Karen Michael, Caleb Principe. Thanks are also due to Raghu Srinivasa for implementing the arithmetic coder.

References

[1] *Advanced Orbiting Systems, Networks and Data Links: Architectural Specification* CCSDS 701.0-B-1, Oct. 1989. Availabe from: CCSDS Secretariat, Communications and Data Systems Division (code-TS), National Aeronautics and Space Administration, Washington, DC 20546.

[2] W. Tai, "Data compression - the end-to-end information systems perspective for NASA space science missions," *Proc. of the Space and Earth Science Data Compression Workshop*, Snowbird, Utah, 1991.

[3] *Advanced Orbiting Systems, Networks and Data Links: summary of concept, rationale and performance* CCSDS 700.0-G-2, Oct. 1989.

[4] P. S. Yeh and W. H. Miller, "The implementation of a lossless data compression module in an advanced orbiting system: issues and tradeoffs," presented at First ESA Workshop on Computer Vision and Onboard Processing, The Netherlands, 1991.

[5] R. F. Rice and J. R. Plaunt, "Adaptive variable-length coding for efficient compression of spacecraft television data," *IEEE Trans. on Communication Technology*, Vol. COM-19, No. 6, pp. 889-897, 1971.

[6] R. F. Rice, "Some practical universal noiseless coding techniques," JPL Publication 79-22, Jet Propulsion Laboratory, Pasadena, California, 1979. Available from: JPL Publication Office, Jet Propulsion Laboratory, Mail Stop 111-130, Pasadena, CAlifornia 91109.

[7] R. F. Rice and J.-J. Lee, "Some practical universal noiseless coding techniques, part II," JPL Publication 83-17, Jet Propulsion Laboratory, Pasadena, California, 1983.

[8] R. F. Rice, P. S. Yeh and W. H. Miller, "Algorithms for a very high speed universal noiseless coding module," JPL Publication 91-1, Jet Propulsion Laboratory, Pasadena, California, 1991.

[9] R. F. Rice, "Practical universal noiseless coding techniques, part III," JPL Publication 91-3, Jet Propulsion Laboratory, Pasadena, California, 1991.

[10] P. S. Yeh, R. F. Rice and W. H. Miller, " On the optimality of code options for a universal noiseless coder," *JPL Pub. 91-2*, 1991.

[11] T. A. Welch, "A technique for high-performance data compression," *IEEE Computer* Vol. 16, No. 6 June 1984.

[12] I. H. Witten, R. M. Neal and J. G. Cleary, "Arithmetic coding for data compression," *Communications of the ACM,* V. 30, N. 6, June 1987.

[13] J. Venbrux, N. Liu, K. Liu, P. Vincent and R. Merrel, "A very high speed lossless compression/decompression chip set," JPL Publication 91-13, Jet Propulsion Laboratory, Pasadena, California, 1991.

[14] J. Venbrux, P. S. Yeh and N. Liu, "A VLSI chip set for high speed lossless data compression," accepted for publication in *IEEE Trans. on Circuits and Systems for Video Technology*, 1992.

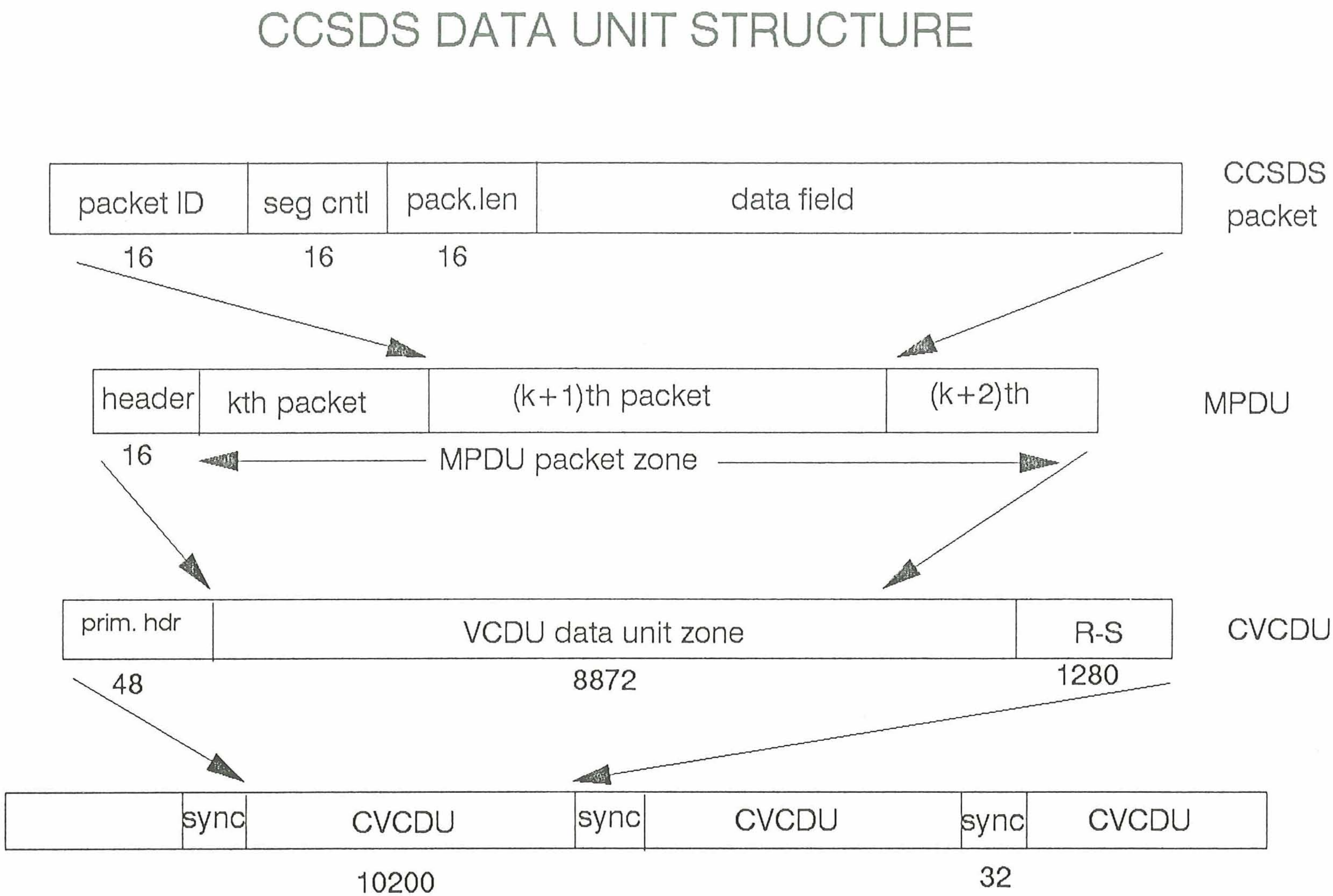


Fig.1 CCSDS packet data structure

C-2

88

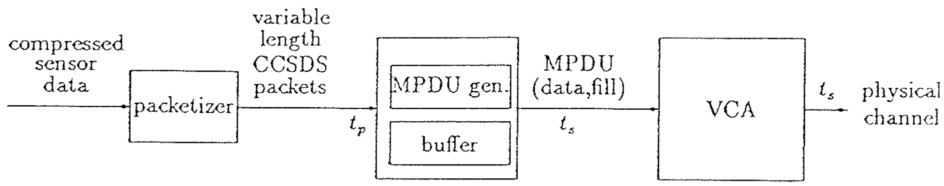


Fig.2(a) Synchronous packet data flow

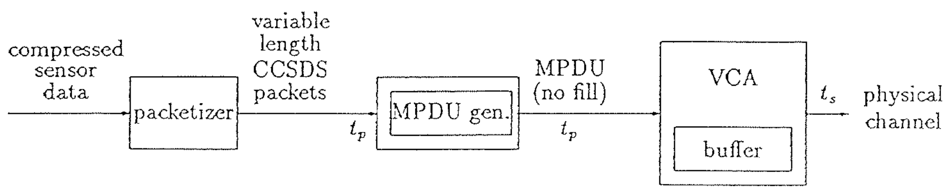


Fig.2(b) Asynchronous packet data flow

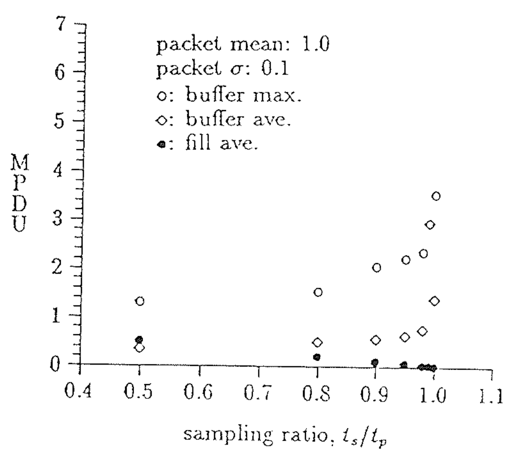


Fig.3(a) Performance of 2(a)

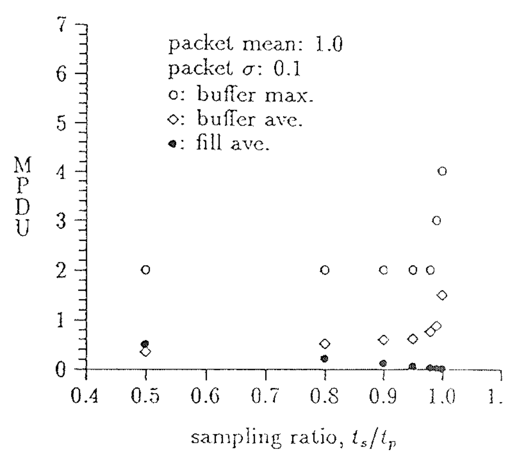


Fig.3(b) Performance of 2(b)

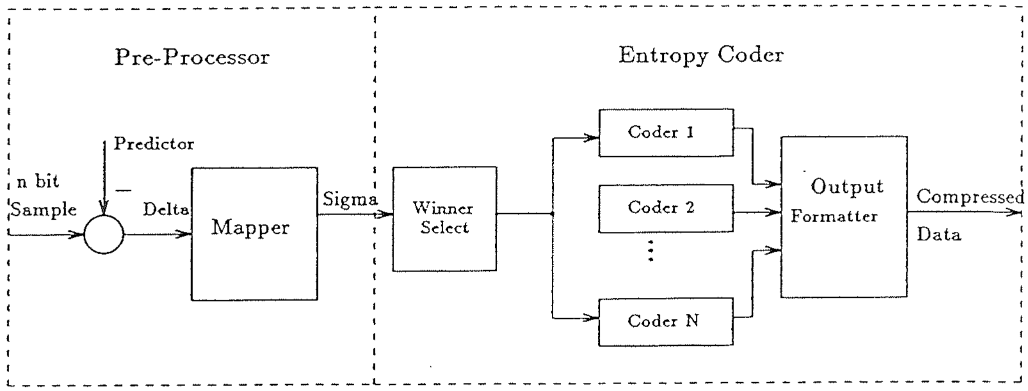


Fig.4 The Rice algorithm architecture

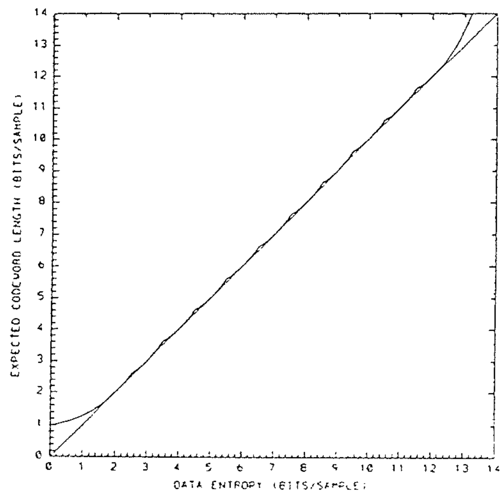


Fig.5 Theoretical performance of the split-sample options

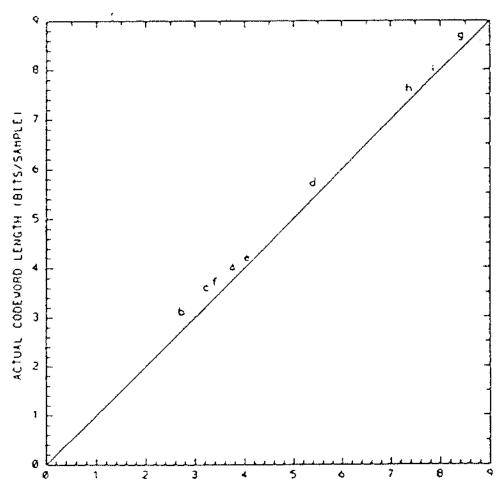


Fig.6 Performance of the coder on samples of 9 aerial imagery

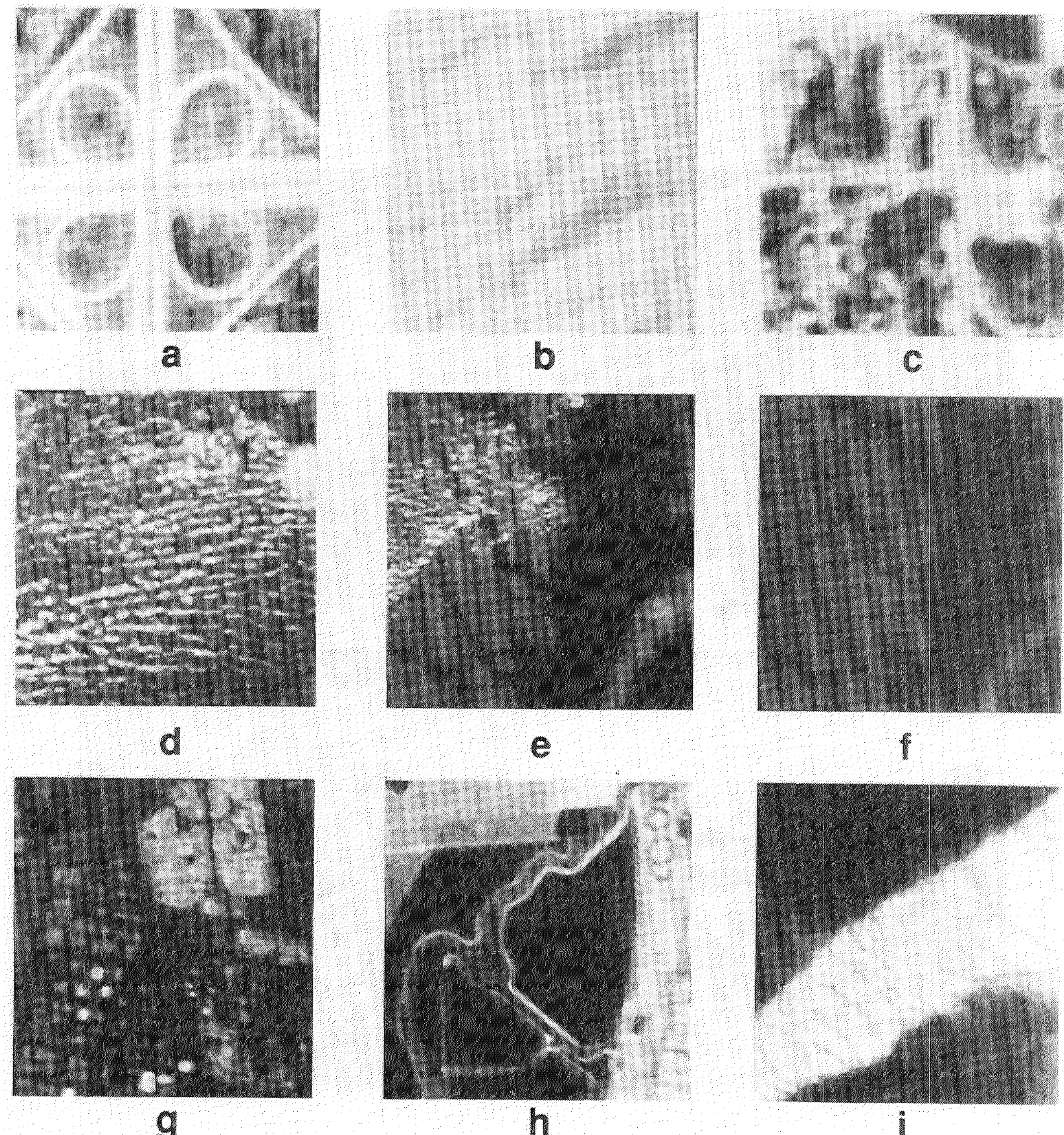
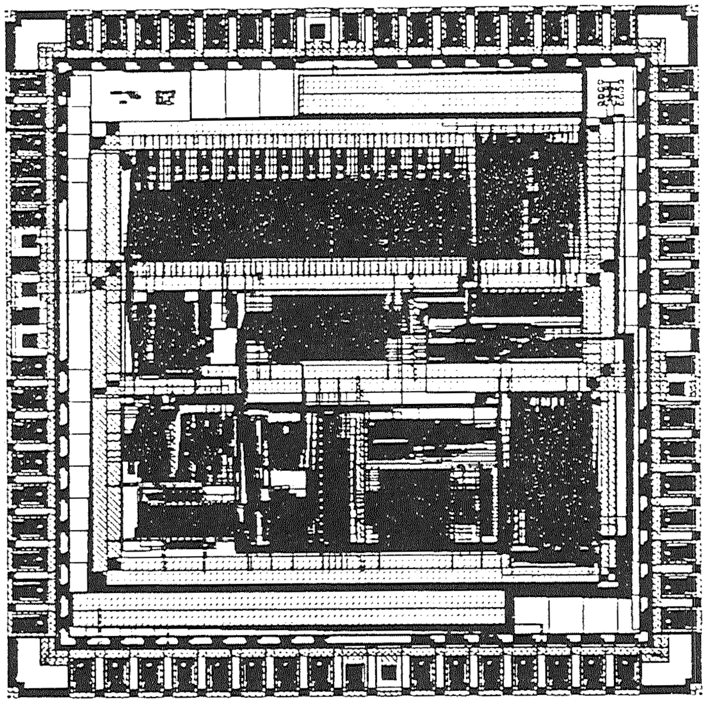
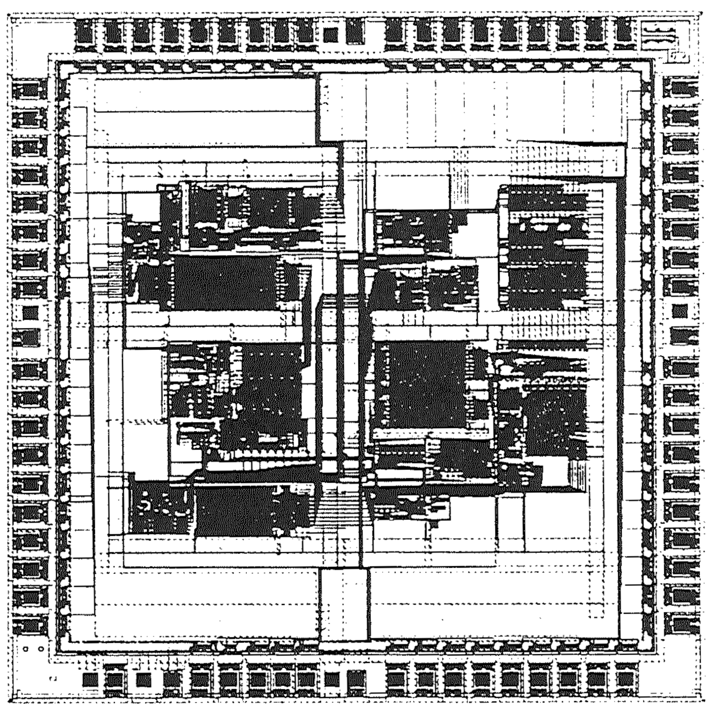


Fig. 7 The 9 GSFC test aerial images

Fig.8 The 4 USC test images



(a) Encoder Layout.



(b) Decoder Layout.

Fig. 9 The chip set layout

	LZ	AH	ARi	p+LZ	p+AH	p+ARi	Rice
girl	28.48	18.80	21.50	30.02	39.40	40.23	41.50
baboon	-4.79	5.90	7.75	-0.15	17.80	18.76	18.44
F16	22.32	14.50	20.76	32.04	40.20	41.65	42.36
aerial	9.42	12.10	13.72	17.63	29.10	29.63	29.95

Table 1: Percentage reduction of files after compression

	LZ	AH	ARi	p+LZ	p+AH	p+ARi	Rice
girl	1.40	1.23	1.27	1.43	1.65	1.67	1.71
baboon	0.95	1.06	1.08	1.00	1.22	1.23	1.23
F16	1.29	1.17	1.26	1.47	1.67	1.71	1.74
aerial	1.10	1.14	1.16	1.21	1.41	1.42	1.43

Table 2: Compression ratio of each test image

	LZ	AH	ARi	p+LZ	p+AH	p+ARi	Rice
girl	46867	53202	51443	45857	39724	39178	38339
baboon	-	246587	241760	-	215575	212976	213816
F16	203621	224052	207697	178132	156865	152945	151091
aerial	237443	230318	226247	215920	185882	184455	183639

Table 3: Total number of coding bytes after compression

	Encoder	**Decoder**
Designed Frequency	20 Mhz	20 Mhz
Design Specs (wc process)	125C,4.5V	70C,4.75V
Measured Lab Freq	50+ Mhz	50 Mhz
Lab Bit Rate N=14	700+ Mbits	350 Mbits
Power(20Mhz,5.5V,100pF+)	0.34 W	0.24 W
Transistors	36,487	33,451
Die Size	5mm X 5mm	5mm X 5mm
Process	1.0um CMOS	1.0uM CMOS
Package	84 pin PLCC	84 pin PLCC

Table 4: Chip Set Summary