TELETEXT DATA CHANGE DETECTION AND NOISELESS DATA COMPRESSION

Y. M. Siu and C. K. Chan
Department of Electronic Engineering
City University of Hong Kong
Tat Chee Avenue
Kowloon, Hong Kong

K. L. Ho
Department of Electrical & Electronic Engineering
University of Hong Kong
Pokfulam Road, Hong Kong

Abstract

Full Channel teletext system is a high speed data broadcasting system. Pages of information are broadcast in a cyclic manner. The detection of data change in the information pages is necessary for data analysis, database update and retransmission. Lossless data compression is also necessary to enhance the data throughput in rebroadcasting and to reduce the storage requirement. Performing data change detection and data compression in real time using a software approach in a small machine is impossible for such high speed data. In this paper, we describe the algorithms that are suitable for hardware implementation for both data change detection and noiseless data compression.

1. Introduction

Full channel teletext is a one-way data broadcasting system [1]. A one way broadcast system is attractive because it may support an unlimited number of users [2]. Pages of information are broadcast in a cyclic manner. The advantage of this system is the high data transmission rate of 6.923 Mbit/s or over 600 pages per second [1]. The disadvantage of these systems are that the service is only suitable for local operation. This is because Teletext is broadcast using either RF or a dedicated cable network; both methods are not suitable for rebroadcasting. The first method consumes valuable spectrum and the latter method is expensive. Also, users may have to wait for a few seconds until the required page arrives. This may be defined as access delay or system response time [2]. In addition, when a user is viewing a particular page, the updated information of the page may only be received when that page is transmitted again at the next cycle. The shortening of update delay is important especially for real time financial data [6]. To extend the coverage, teletext data may be rebroadcast using telephone lines or data lines to anywhere in the world as reported in [3] or bridged to different networks [4]. The access delay may be improved by storing the entire database in the local memory of the PC/terminal [3][5][6]. The shortening of the update delay may be reduced by sending the changed or updated information using a small secondary channel [6]. In [6], a slow data rate VBI teletext which does not require extra spectrum is used to broadcast real time financial data. The slower data rate of VBI teletext as compared with full channel teletext is compensated by using the ghost rows together with the storage capabilities of modern terminals to shorten both the access time and the update delay [6]. Since VBI teletext does not require extra spectrum, it may also be used to rebroadcast full channel teletext.

Due to the limited bandwidth of the telephone network and the VBI teletext system, the methods proposed in [3][6] only transmit the changed or updated data in the telephone lines and ghost rows. Therefore, these methods require the detection of data change in the original full channel system. Also, the data should be compressed before transmission to enhance the data throughput.

It is however impossible to use software in a small machine to detect all changes and perform the data compression in real time due to the high data rate. It would require a lot of processing power to

Contributed Paper
Manuscript received May 24, 1995

0098 3063/95 $04.00 © 1995 IEEE

check over 600 pages of information per second and this is beyond the capability of a small machine. In this paper, we propose to use hardware to perform both data change detection and data compression so that the system server/transmitter can be implemented using a PC.

2. Hardware Data Change Detection Algorithm

The design is based on the fact that CRC (cyclic redundancy check) is capable of burst multiple bit error detection, and is widely used to detect errors in received serial data streams. We demonstrate in this paper that it may also be used for data change detection. This CRC method is also used in the string matching comparison in the compression encoding process as described in section 3. The detector is implemented as a PC card. The teletext signal, after decoding, is stored in the RAM in the format of 8-bit alphamosaic characters for display [7]. A standard teletext decoder (e.g. SAA5231, SAA5243) is programmed to receive all available pages. While the decoded data are being written into the RAM, the relevant data are directed to a hardware CRC generator to compute a code for every page. These codes are compared with the corresponding codes computed in the previous cycle. Any changes in the page content will result in a different code being computed and thus detected.

2.1 Analysis of Data Change Detection Using CRC

The hardware CRC generator is constructed of delays (flip-flops), exclusive-ORs and feedback lines[8].

Analysis of data change detection using this method may be performed by relating them to a polynomial in which the coefficients contained in the CRC generator are associated with the polynomial $p(x)$ of degree r.
If the incoming binary data stream is represented as a polynomial $m(x)$ of degree n.

$$m(x) = q(x).p(x) + r(x) \tag{1}$$

This may be implemented using flip-flops and exclusive-OR gates [8]. After n shifts, the remainder $r(x)$ is used as a signature. It is useful to the extent that different data streams generate different signatures.

Data change patterns only go undetected if the change pattern is an integer factor times the generator polynomial. If we assume random patterns, $n+r$ bits produce a total of 2^{n+r} possible patterns. The number of integer multiples of a generator polynomial of degree r in a sequence of length $n+r=2^n$. Each multiple can be considered as a finite sum of n factors.

Therefore, the probability of undetected change pattern $= \frac{2^n}{2^{n+r}} = \frac{1}{2^r}$ (2)

Data pattern Change Detected Percentage

$$= (1-\frac{1}{2^r})x100\% \tag{3}$$

when r = 16 Detection rate = 99.998%
r = 32 Detection rate = 99.999%

which is extremely high and is the same as using CRC for error detection.

The particular changed pattern that may be undetectable is determined by the feedback coefficients and hence by the value chosen for the divisor. In practical implementations of signature analysis, the choice of feedback coefficients is governed by two main considerations :
The free-running feedback shift register should be made to cycle through all possible state. i.e. maximum cycle length. It is always possible to choose the feedback coefficients so as to achieve maximal length cycle [9][10]. Regularly spaced feedback taps should be avoided, particularly four- or eight- bit spacing [10].

In our system, a 16-bit CRC generator with polynomial $p(x) = x^{16} + x^{12} + x^5 + 1$ is used and the result is satisfactory.

2.2 Data Change Detector Implementation

The decoded teletext page number and data are always stored in the same locations in the RAM [7]. These are latched from the data bus of the

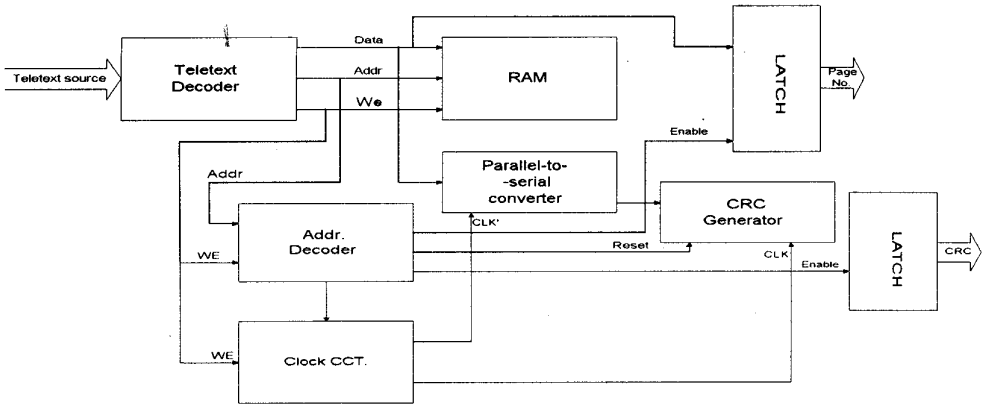


Fig. 1 Hardware Data Change Detector

teletext decoder. The control signals and timing are obtained from the WRITE (WE) signal and the address bus (Fig. 1). An address decoder is used to control the latches to gate the page number and the CRC code. It also enables the clock circuit to start the CRC computation for the relevant data. At the beginning of the page, an output from the address decoder will reset the CRC generator. A parallel-to-serial converter is used to convert the 8-bit data into serial data before they are fed into the CRC generator.

The synchronisation of the clock circuit is controlled by the WRITE signal (WE) of the teletext decoder.

A counter, which is not shown in Fig. 1, is included in the clock circuit to ensure that only 8 clock pulses are generated for every character received. In addition to the detection block shown in Fig. 1, a microcontroller is used to compare the CRC codes generated in this cycle and the previous cycle. At the end of a teletext page reception, an interrupt to the microcontroller would be generated by the address decoder to read the page number and the CRC from the latches. The compared results are passed on to the PC. When a teletext page with changes in information is found, the PC will then retrieve that page and update the database for local viewing or rebroadcasting.

3. Noiseless Data Compression

Data compression is important for data transmission, especially for transmitting data using narrow bandwidth channels as mentioned above. The compression algorithm described in this paper is suitable for system broadcasting data in English text format, such as the full channel teletext system of the Stock Exchange of Hong Kong. Noiseless text compression algorithms can be divided into two classes: statistical and dictionary. Dictionary methods do not compress as well as statistical methods, but they are fast and easy to implement. They only use a modest amount of computer time and memory, especially on the decoder side. This is important for data broadcasting systems which use one encoder/transmitters and many decoders/receivers hierarchy and therefore, the decoders have to be simple.

In general, the data structure of a dictionary allows the storage of a set S of N distinct elements in a word table. Each element and its associated records can be retrieved using a key k. On the encoder side, when an element x is presented, the problem is to find whether this element is stored in the table and where it is located. (i.e. the key). After the search is completed, it is either successful, having generated the key for that element, or it is unsuccessful, having determined that the key is nowhere to be found (i.e. the element is not stored in the table). The key k or the actual element (if no key can be found) is then transmitted to the decoders. The decoder either receives the element directly or retrieves the element simply by using the received key.

There are different algorithms to generate and search for these keys. Trie search is a fast searching method with no collision problems, but it requires a large amount of memories [11]. Hashing is an efficient method to generate the required key which promises fast access. However, most hashing methods involve a certain amount of wasted time due to the need to resolve collisions [11][12]. This includes the perfect hashing function, because a hash function which is perfect for one key set (cause no collision) may not be perfect for another key set.

In our system, the hardware search algorithm is based on the algorithm proposed by Pearson [13]. The hashing function takes a word W to be compressed which consists of a number of n characters $C_1C_2C_3...C_n$ and uses an auxiliary table T of 256 randomise bytes in the following hashing process.

```
h[0] := 0;
for i in 1...n loop
        h[i] := T[h[i-1] xor c_i ];
end loop;
return h[n];
```

Since words are separated by a space character in English text, the looping operation may be terminated when a space character is detected. Therefore, words can be of variable length. In [13], this hashing function can return 256 addresses and therefore the size of the word table is 256 words. It seems that such a small word table may not be adequate for normal text message. However, since full channel teletext is normally used to broadcast specific information such as financial data, the table contents may be arranged so that different tables can be used for different types of broadcast information in the system.

In our system, we enhance Pearson's method by having 16 word tables, each with fixed word length ranging from 1 to 16 characters. This arrangement increases the number of words available. Also, the collision of two useful words in a fixed length table can be reduced by using the good separation property of the hashing function for words of the same length [13].

3.1 Word Table Construction

The compression performance of all dictionary-based compression algorithms is highly dependent on the probability of finding the words in the word tables. Therefore, the initialisation of word tables should be based on the statistical analysis of the message. Useful words are assigned to their corresponding locations according to the address generated by the hashing function by using the same T table. If collision occurs, the lower priority word should be discarded. However, if the higher priority word is composed of a prefix word appended with one or few suffix characters and the prefix part has already assigned to a lower order table, this word should be discarded instead. Each word in the word tables has an associated CRC of that word. This is used to simplify the word matching hardware in the encoding process. With this method, the word matching comparison only requires a 16 bit comparitor instead of having to compare all the characters in the words.

3.2 Encoding Algorithm

For normal English text message, words are separated by spaces which are used as an "end of word" symbol. It is not necessary to transmit the space character and the decoder will append one after a word is decoded. The encoding algorithm divides words into two classes, the copy words and the literal words. Copy words are those that can be found in the word tables. Literal words are words that cannot be found in the word tables and therefore cannot be compressed. For word length that is longer than the maximum table order of 16, the cascading word ID will be used. If words are separated by more than one space, the first space character after the encoded word will not be sent as mentioned above. The other space characters will be treated as a word and the length of it is determined until a non-space character is encountered. The copy word and the literal word formats are shown in Fig. 2.

The copy word format

2 bits	4 bits	8 bits
Word ID	Table Order	Table address

word ID 00 : Copy word
10 : Cascading-Copy word

The literal word format

2 bits	4 bits	8 bits		
Word ID	Word Length	ASCII Code of Character	. . .	ASCII Code of Character

word ID 01 : Copy word
11 : Cascading-Literal word

Fig. 2 Copy Word And Literal Word Format

3.3 Compression Performance

The compression performance may be measured by the compression ratio. It may be defined as follows :

(CR) = bit length of word/bit length of encoded word

Since we categorise words into different character length, there are different compression ratios for words of different character length. It should be noted that when calculating the bit length of a word, the space character that follows should also be included. Each copy word may be represented by 14 bits (Fig.2). Therefore, the compression ratio of a n character copy word is $CR_n(c)$.

$$CR_n(c) = \frac{(n+1)\cdot 8}{14} \tag{4}$$

For literal words, since the redundant 6-bit header is absorbed by the space character following the word, the compression ratio is

$$CR_n(l) = \frac{(n+1)\cdot 8}{6+8n} \tag{5}$$

Therefore the algorithm will not expand the original data even if no matching is found. But for words longer than 16 characters, one extra cascading header is required to inform the decoder not to append a space character after the first 16 encoded prefix characters. However, words of length longer than 16 characters are very exceptional and unlikely to happen very often.

Let the probability of finding a matched word of n character length in the order-n word table be p_n.

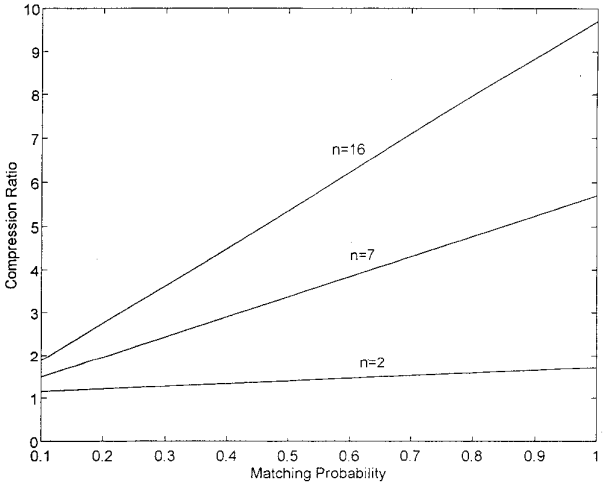
The compression ratio for a n character word having matching probability p_n is :

$$CR_n(P_n) = P_n \cdot CR_n(c) + (1-P_n)\cdot CR_n(l) \tag{6}$$

The matching probability p_n depends on the correspondence of the word tables to the message. Therefore, different tables may be used for different types of broadcast data to increase the matching probability

Fig. 3 shows the compression ratio versus matching probability for different character length words.

Fig. 3 Compression Ratio Versus Matching Probability



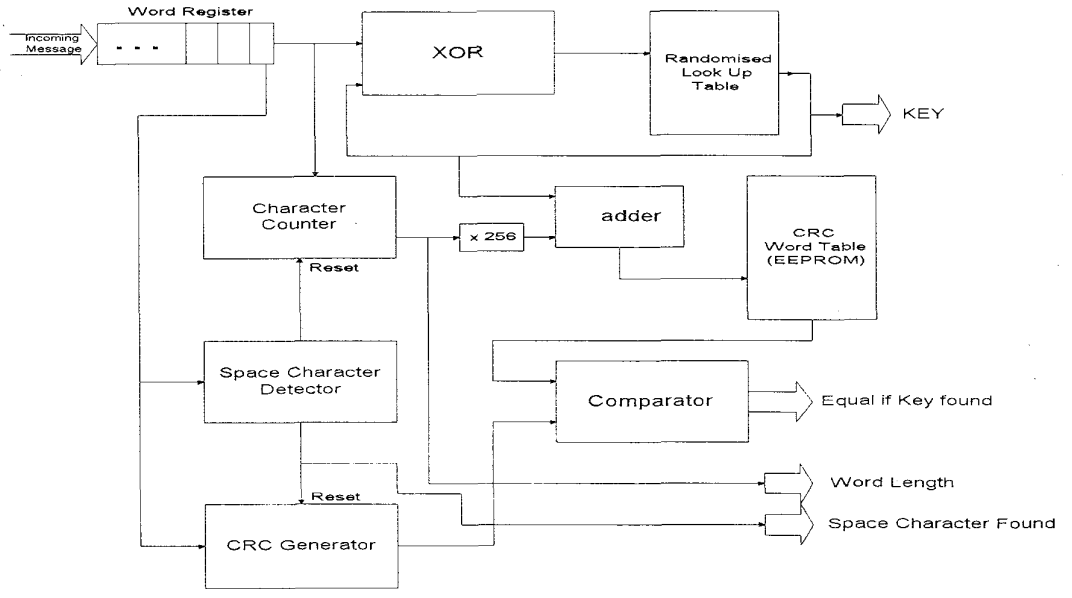


Fig. 4 Hardware Data Compressor

Let $D_n(P_n)$ be the distribution function of the matching probability for n character words in a message. The weighted mean of $\overline{CR_n}$ is :

$$\overline{CR_n} = \sum_{P_n=0}^{P_n=1} D_n(P_n) \cdot CR_n(P_n) \tag{7}$$

Let W_n be the probability of occurrence of n character words in the message to be compressed. The average compression ratio of the message is $CR = \sum W_n \cdot \overline{CR_n}$

$$CR = \sum_{n=1}^{n=l} W_n \cdot \sum_{P_n=0}^{P_{n=1}} D_n(P_n) \cdot CR_n(P_n) \tag{8}$$

Where

P_n = matching probability for n character words.

$CR_n(P_n)$ = compression ratio for n character words with matching probability P_n.

$D_n(P_n)$ = distribution function of n character words with matching probability p_n

W_n = probability of occurrence of n character word in the message

l = the maximum character length word in the message

From equation (8), it can be observed that the compression ratio of the algorithm is dependent on several parameters. The parameters W_n and l are dependent on the message which we have no control over. The parameter $CR_n(P_n)$ is determined by the algorithm. The parameter $D_n(P_n)$ depends on the correspondence of the word tables to the message to be compressed. These parameters affect the performance of this dictionary based compression algorithm. Different sets of word tables may be used for different types of broadcast messages as mentioned above.

3.4 Compression Encoding Algorithm Implementation

The compression algorithm is implemented as shown in Fig 4. The randomised hash table is stored in a ROM, the word tables which store the corresponding CRC of the words are constructed using an EEPROM. Different word tables may be programmed into the EEPROM for systems broadcasting different types of data as mentioned above. Initially, the address of the hash table and the character counter is reset to 0. Each character of the word is shifted into the register in turn and the character counter is incremented by 1. Each character is XOR with the data from the hash table

and the result is used as the next address of the hash table. The incoming character is also fed to the CRC generator. When a space character is detected, it means a word is finished. The output of the character counter is multiplied by 256 to point to the appropriate table and the output of the hash table (the key) is used as the offset to point to the corresponding word of that table via an adder. The output of the word table, the CRC of the corresponding word is used to compare with the CRC generated from the incoming word. If both are the same, a key is found, otherwise there is no compression for this word. The character counter, the address to the hash table and the CRC generator is then reset for the next incoming word.

4. Conclusions

We have illustrated the use of CRC and dictionary based compression scheme using a hashing function to detect changes and compression respectively in teletext pages. Both data change detection and compression encoding process can be implemented on an add on card in a PC without using any specialised components. They are suitable for detecting changes and data compression in real time for teletext systems. There is no need to have additional hardware for the decoding process since it is only a lookup table operation. This is extremely suitable for rebroadcast systems based on the one transmitter and many receivers hierarchy

5. Acknowledgement

Some of the analysis work of the compression algorithm was carried out by Mr. Wah-King Tam. Mr. Tam was a MSc student in City University of Hong Kong (93-94).

6. References

[1] Dennis N. Pim: 'Television and Teletext', Macmillan Education

[2] John W. Wong, "Broadcast Delivery", Proceedings of the IEEE, Vol. 76, No. 12, December 1988.

[3] Y.M. Siu and C.K. Chan: 'Rebroadcasting of Video Teletext Data Through the Telephone Network - Design and Implementation'. 1991 IEEE Workshop on Visual Signal Processing and Communications, pp 217-221

[4] H.K. Pung and Tan Boon Tiong: 'The Design, Implementation and Performance Measurement of a Teletext Server For A Ethernet', Proceedings of International Conference On Communication Systems ICCS'90, Singapore, pp 26.7.1-26.7.5.

[5] M. Gunduzalp, "Downloading All Teletext Pages To a Computer", IEEE Transactions on Consumer Electronics, Vol.39, No. 4, November 1993, pp 832-836.

[6] Y.M. Siu, C.K. Chan and K.L. Ho, "Financial Data Broadcasting Using VBI Teletext", 1994 International Symposium on Consumer Electronics, Hong Kong, Nov. 1994, pp 190-196.

[7] J.R. Kinghorn: Philips Components Report No. MTV89008 'Enhanced Computer Controlled Teletext SAA5243 Series User Manual'

[8] G. Albertengo & R. Sisto, "Parallel CRC Generation", IEEE Micro, October 1990, pp 63-71.

[9] R.L. Pickholtz, D.L. Schiling & L.B. Milstein, "Theory of Spread-Spectrum Communications - A Tutorial", IEEE Transactions On Communications, VOL.COM-30, NO. 5, May 1982, pp 855-884.

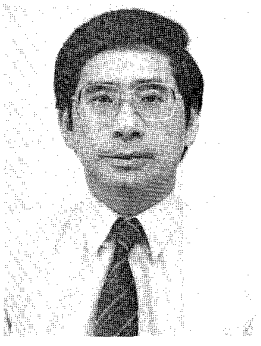
[10] S.W. Golomb,Shift Register Sequences. San Francisco, CA: Holden Day, 1967.

[11] E.A.Fox, L.S. Heath, "Practical Minimal Perfect Hash Functions For Large Databases", Communications of The ACM, January 1992, Vol.35, No.1, pp 105-121.

[12] M.D. Brain & A.L. Tharp, "Using Tries to Eliminate Pattern Collisions in Perfect Hashing", IEEE Transactions on Knowledge and Data Engineering, VOL.6, No.2, April 1994, pp239-247.

[13] P.K. Pearson, "Fast Hashing of Variable Length Text Strings", Communications of theACM, June 1990, Vol.33, No.6, pp677-680.

Yun Ming Siu received the B.Sc. degree in 1981 from the University of Manchester, UK. From 1981-1989, he worked at Racal-BCC, UK, first as a development Engineer, and then as a team leader responsible for the development of frequency hopping radio and Tactical communications systems. In 1990, he joined the Department of Electronic Engineering at the City University of Hong Kong as a Lecturer. His current research interests include information broadcasting and delivery systems, and communication networks. He is a Chartered Engineer of the Engineering Council, UK, a member of the Institution of Electrical Engineers, UK, Chinese Institution of Electronics and the Hong Kong Institution of Engineers.



Ka Leung Ho received the B.Sc.(Eng.) degree and the M.Phil. degree in Electrical Engineering from the University of Hong Kong in 1971 and 1973, respectively, and the Ph.D. degree from the University of London in 1977. In 1984, he joined the Department of Electrical and Electronic Engineering at the University of Hong Kong and is currently a senior lecturer. His current research interests include millimetre wave propagation, signal processing and communication systems. Dr. Ho is a member of IEEE.

Chok Ki CHAN received the B.S. and M.S. degrees both in E.E. from University of California, Los Angeles in 1977 and 1978 respectively. He received the Ph.D. degree in Electronics from the Chinese University of Hong Kong in 1984. In 1978 and 1979, he was with the Severe Environment Systems Co. Ltd., California, as a design engineer of computer systems. From 1980 to 1985, he was with the faculty of the Department of Electronic Engineering, Hong Kong Polytechnic. He is currently a Reader in the Department of Electronic Engineering, City University of Hong Kong. His current research interests include image compression, signal processing and information delivery. Dr. Chan is a Chartered Engineer of the Engineering Council, U.K. and a member of the Institution of Electrical Engineers, U.K., and the Hong Kong Institution of Engineers, Hong Kong. Currently, he is the Vice Chairman of the IEEE Hong Kong Section.