Implementation of LZW Data Lossless Compression Algorithm Based on VB

Yuan Qinghui[1, a,*], Nie Xiujun[1, b] and Yuan Qingfei [2,c]

[1]Binzhou Polytechnic, Binzhou, Shandong Province China,256603

[2]China Offshore Bitumen Co.LTD, Binzhou, Shandong Province China,256600

[a]sdyqh1979@126.com, [b]443183538@qq.com, [c]yuanqingfei2007@126.com

Keywords: Data compression, LZW algorithm, Lossless compression, VB

Abstract: The first analysis of this paper chooses the LZW compression algorithm for data compression, it is the basic principle adopted a series of advanced compression table, and compressed file only digital storage, rather than storing strings to achieve the purpose of data compression. In this paper, VB programming language, Visual basic 6.0 software through the design of the platform design LZW data compression software, has been tested, to achieve the objective of the data compression.

Introduction

In recent years, along with the computer technology, network technology and multimedia communication technology rapid development, The amount of data in each system is more and more big, which brings serious obstacles to the storage of the data, transmission, and effective and fast access to information. So the data compression technology is the key technology to solve this problem[1].

Data compression technology has a very important role in the rapid development of computer technology today. In addition to reducing the system overhead, data compression technology has another advantage. When the data compression is used to reduce the storage space, the overall execution time of the program can be reduced. When data compression is used for data transmission, the transmission character is less, which can reduce the probability of error in the transmission data, and improve the efficiency of the data transmission.

System working principle

Overview of data compression and the principle of LZW algorithm

Data compression is signal from source made with the minimum number of digital signals, and it can reduce the signal space of a given message set or collection of data. The theory research about data compression has began on the basis of the Shannon information theory. When Shannon information theory is founded, he put forward the data as a combination of information and redundancy. Therefore, In simple terms compression is that long data is replaced with short code ,and the redundancy is reduced to a minimum, reduce duplication of information in the file in order to achieve the purpose of data compression. Data compression according to the reversibility of the compression process can be divided into lossy compression and lossless compression. Lossless compression used statistical redundancy of data to compress, after decompression the original data can be fully restored to without causing any distortion, but the compression rate is limited by the statistical redundancy of the data, which is generally 2:1 to 5:1. Typical lossless compression coding include: Hoffman coding, arithmetic coding, stroke coding, LZW coding, etc..

In this paper, we mainly study the LZW compression algorithm in lossless data compression technology. LZW algorithm uses an advanced string table compression, the basic principle of it is as follows：above all a string table is established, in which each of first character string appeared is put ,and it is used a number to represent. This number is related to the position of the string in the string table, when the string appears again, that is, it can be used to represent its number, and this number is stored in the file. In decompression, the string table can be re generated according to the compressed data.

Main working principle

In this paper, we use VB language programming to achieve data compression, compiler software using basic Visual 6. Functional block diagram is shown in figure 1. The software makes the data to be processed through the software compression module for compression processing to generate new data, and then compares the new data with the original data, analyzes compression ratio; Compression data recoveries through the software decompression module for decompression processing, then checks the data and the original data recovery ,and views the error analysis results.

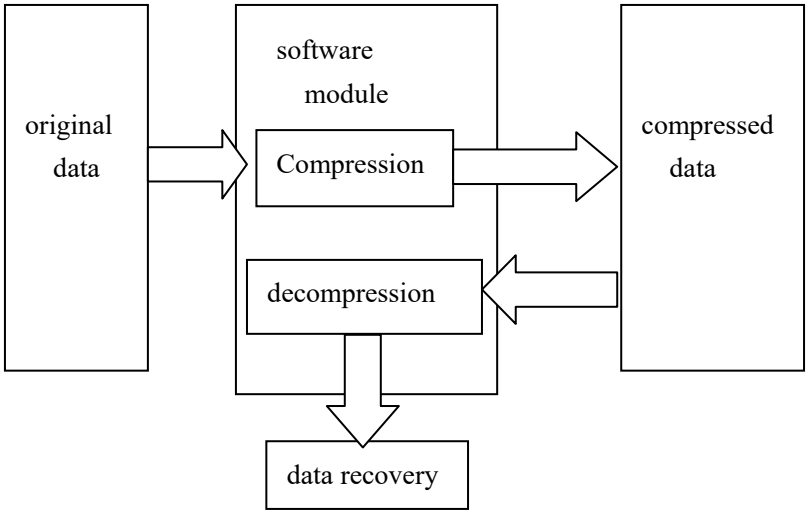

Fig. 1 Functional block diagram

Software compression module and hardware circuit module

(1)Data compression flow chart

Data compression flow chart is shown in figure 2

The related characters are explained as follows:

tab_size: Hash table length; codevalue()：dictionary table；

precode(): prefix code table; appendcode()：suffix code table；

nextcode: code; maxcode: maximum dictionary encoding; index: dictionary index；

offset：offset (When a conflict occurs, offset is used as intermediate volume the used to resolve conflicts)；

p，c：input byte；256：Compressed data start flag；257：Compression data end flag.

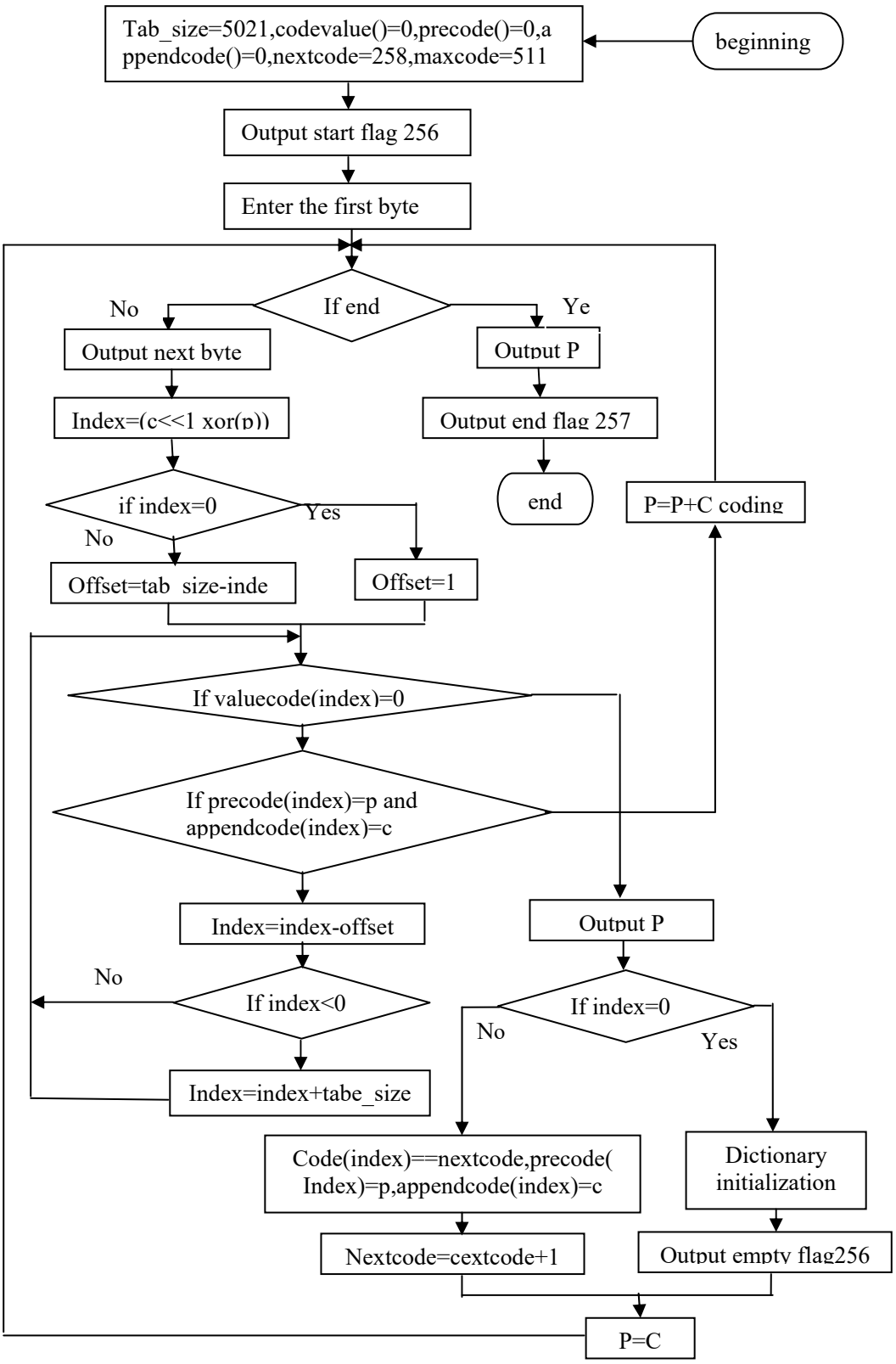

Fig. 2 Compression flow chart

(2) VB procedures and code description

The full name of VB is Basic Visual, which is a Windows application development tool introduced by Microsoft company. VB is one of the most widely used programming languages in the world, and it is widely recognized as the most efficient programming

method in the world. As long as beginners have mastered a few key words, they can establish a practical application. [2]。

```
Dictionary initialization and hash table lookup module
Private Sub Init()
Dim i As Long
or i = 0 To 255
Dict(i) = Encode(i)
Next
End Sub
Private Function Search(p As Variant, c As Variant) As Integer
Dim a As Integer
Dim index As Integer
Dim offset As Integer
index = c * 2 xor p
If index = 0 Then
offset = 1
Else
offset = 5021 - index
End If
recontinue:
If codevalue(index) = -1 Then
Search = index
GoTo outexit
End If
If prefixcode(index) = p And appendchar(index) = c Then
Search = index
GoTo outexit
End If
index = index - offset
If index < 0 Then
index = index + 5021
GoTo recontinue
End If
outexit:
End Function
```

This part of the procedure is the initialization of the dictionary, above all, the first 256 numbers corresponding to the value of the 0-255 are converted into a binary number. Then we determine the position of the string in the dictionary according to the Hashi location and set the offset，and determine whether the current address code codevalue (index) is empty. If there is a string in the dictionary, we check whether the address conflict, if the address conflict, then we readdress and continue to find.

```
Dictionary reset module
Private Sub Wipe1()
Dim i As Integer
For i = 256 To 511
Dict(i) = -1
Next
Count = 258
End Sub
Private Function Wipe() As Integer
Dim i As Long
For i = 0 To 5021
codevalue(i) = -1
Next
Wipe = 258
End Function
```

When the compressed data encoding is more than 9 bits and the set of Hashi table length exceeds 5021, then this part of the program remove the dictionary and code start again from 258.

```
Dictionary add module
```



```
index = Search(p, c)
If codevalue(index) <> -1 Then
p = codevalue(index)
Else
length = length + 1
cont(length) = p
If nextcode <= 511 Then
codevalue(index) = nextcode
nextcode = nextcode + 1
prefixcode(index) = p
appendchar(index) = c
Else
length = length + 1
cont(length) = 256
nextcode = Wipe
End If
p = c
End If
```

This part of the program is the add program of the dictionary. If greater than, which initialize dictionary and output empty flag 256, otherwise writes Nextcode, P, and C to the dictionary., nextcode= nextcode+1.[3]

(3) LZW algorithm hardware circuit module

Hardware circuit design is the use of VHDL language coding and compiled by QUARTUS software. The integration of each module of the compression algorithm is shown in Figure 3.

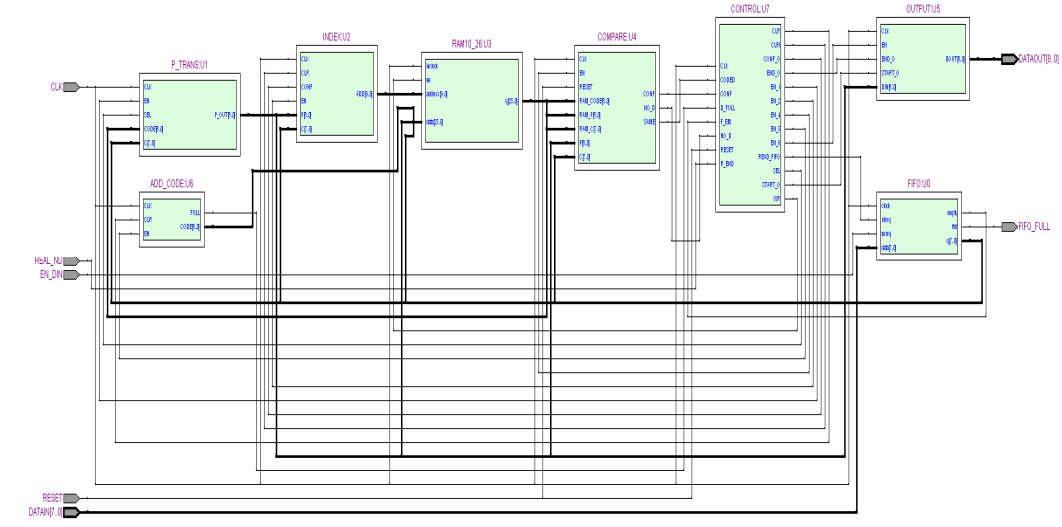

Fig. 3 circuit integration diagram

The function analysis of each module of the schematic diagram:

1)FIFO modular：to buffer the received date, and prevent the input data overflow because the data has not been compressed;

2)CONTRAL modular：The module is a state machine, which controls the work of each module in accordance with the prescribed time sequence;

3)ADD_CODE modular：Realization of the function: nextcode= nextcode+1，Function :check whether the code is greater than 511, if greater than, which initialize dictionary and output empty flag 256, otherwise writes Nextcode, P, and C to the dictionary, nextcode= nextcode+1;

4)P_TRANS modular：Give the next byte to the variable P, that is, P=C; if there is no conflict, We will use one byte to represent the byte P and C ,and truly realize the compression, that is, P = P+C;

5)INDEX modular：Dictionary index determines the position of the string in the dictionary and sets the offset;

6)RAM10_26 modular：LZW compression dictionary storage unit;

7)COPARE modular：Compare: if codevalue (index) is equal to P and appendcode (index) is equal to C, if equal, the P is equal to the encoding obtained by P plus C, otherwise the address conflict, you want to regenerate the dictionary address.

8)OUTPUT modular：output compressed stream. [4]

Software decompression module

Decompression algorithm is just the inverse process of compression algorithms, but also a dynamic generation a string table, and then according to read the code, which compresses data reduction. The input stream is code output stream of compressed algorithm, and the output stream is the input stream of compression algorithm.

(1) Decompression flow chart

The decompression flow chart is shown in Figure 4, in which the related characters are explained as follows:

p, c: input encoding value; 256: compressed data start flag;257: compression data end flag.

(2) Program code description

The dictionary initialization module is the same as the initialization module during compression.

```
Judgment code start and end module
cw = cont1(1)
o = Dict(cw)
If cw = 256 Then
cw = cont1(i)
o = o & Dict(cw)
i = i + 1
GoTo bottom
End If
If cw = 257 Then    GoTo done
```

This part of the program can determine the beginning and end of the coding, When the code is 256, it indicates that the dictionary is full and begins to extract from the next data. When the code is 257, it indicates that the decompression is over.

```
To see if the encoding is present in the dictionary module
If Not Dict(cw) = -1
Then
o = o & Dict(cw)
p = Dict(pw)
c = Mid(Dict(cw), 1, 8)
Add (p & c)
ElseIf Dict(cw) = -1 Then
p = Dict(pw)
c = Mid(Dict(pw), 1, 8)
o = o & p & c
Add (p & c)
End If
```

This part of the program can determine whether the code exist in dictionary, if there is in the dictionary, then output, and put the string and the previous string into the dictionary; If

there is no in the dictionary, output previous string plus 1-8 bit of previous string, and put them in the dictionary.

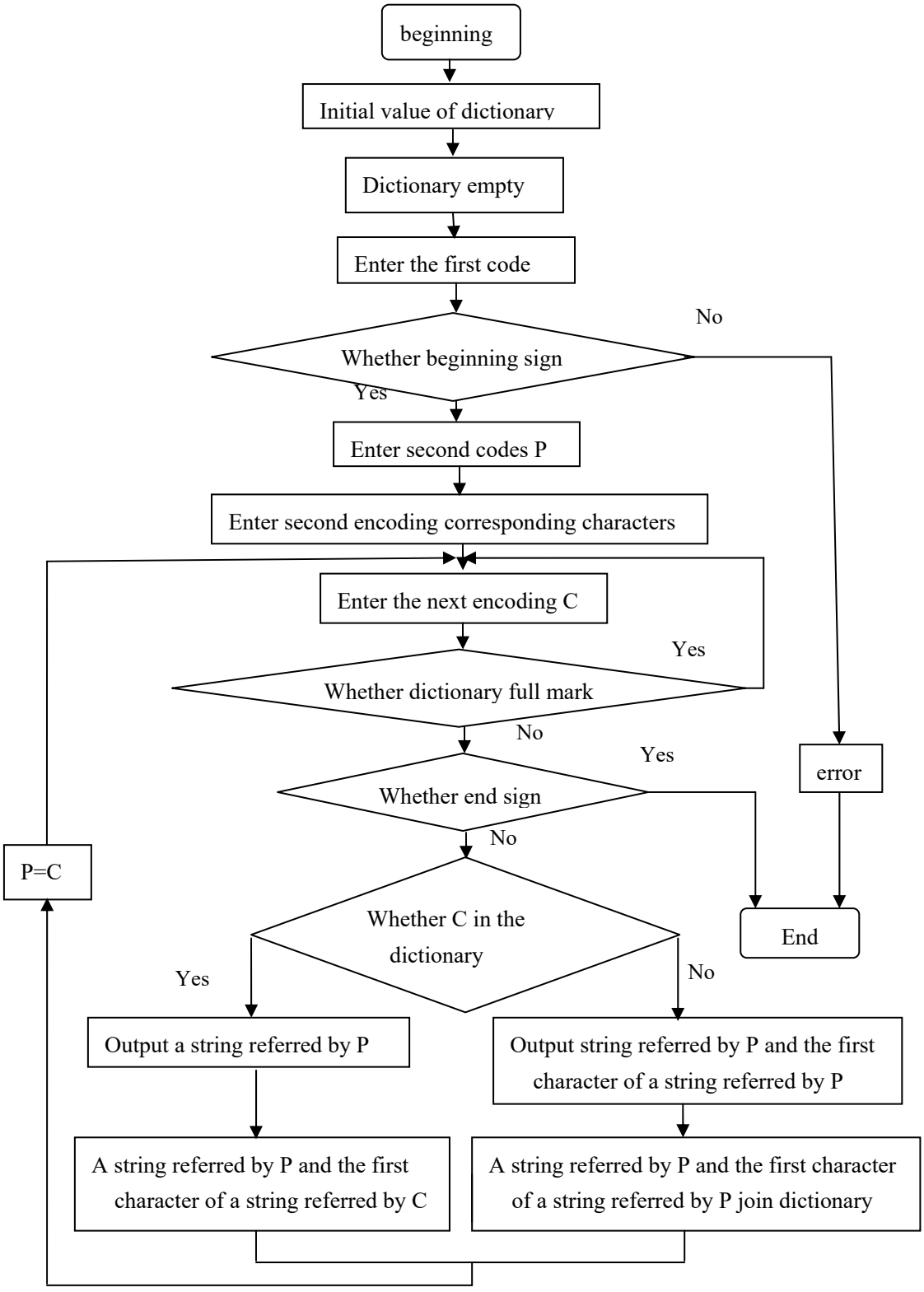

Figure. 4 decompression flow chart

Conclusion

This paper focuses on the analysis of algorithm principle of the LZW compression and decompression and the concrete implementation steps. LZW data compression software designed through the basic Visual 6 software design platform, after testing, it does achieve the purpose of data compression.

LZW algorithm is not only fast, but also it has a better compression effect for all types of computing documents. Knowing the data compression technology, it is the most important and basic method to realize the real-time, fast and effective processing, transmission and storage of data. So information dissemination (such as TV), transmission (such as voice communication) and storage (cloud diagram of computer storage) are inseparable from the data compression, data compression technology also has a wide application prospect.

References

[1] Wang Ping. Realization and research of LZW lossless compression algorithm. computer engineering, Vol. 7(2002), p. 98～99.

[2] Li Jinmin, Zhang Wendong, Mao Haiyang. Research on the implementation of the hardware implementation of lossless data compression algorithm. Journal of Harbin Institute of Technology, Vol. 2(2006), p. 315～316.

[3] Yang Guoliang, Zhang Guanghua. Lossless LZW compression algorithm and its implementation. Journal of Capital Normal University (NATURAL SCIENCE EDITION), Vol. 25(2004), p.12～13.

[4] Jin Weimin. Research on the application of LZW algorithm in data communication [J]. computer engineering and science, Jin Weimin. Research on the application of LZW algorithm in data communication. computer engineering and science,Vol. 26(2004), p.46～48.

Author introduction

Yuan Qinghui, male, Jining Shandong, Binzhou Polytechnic, mainly engaged in electrical automation engineering research and management.

