Huffman Encoding using VLSI

Shilpa.K.Meshram
VLSI
S.R.K.N.E.C,Nagpur.
shilpa_ukey@sify.com,

Meghana .A. Hasamnis
Electronics dept,
S.R.K.N.E.C,Nagpur.
meghanahasamnis@rediffmail.com,

Abstract – Huffman coding is entropy encoding algorithm used for lossless data compression. It basically uses variable length coding which is done using binary tree method. In our implementation of Huffman encoder, more frequent input data is encoded with less number of binary bits than the data with less frequency.This way of coding is used in JPEG and MPEG for image compression. Huffman coding uses a specific method for choosing the representation for each symbol, resulting in a prefix code. Prefix-free codes means the bit string representing some particular symbol is never a prefix of the bit string representing any other symbol.

Keywords-Compession, Binary tree, Histogram.

I. INTRODUCTION

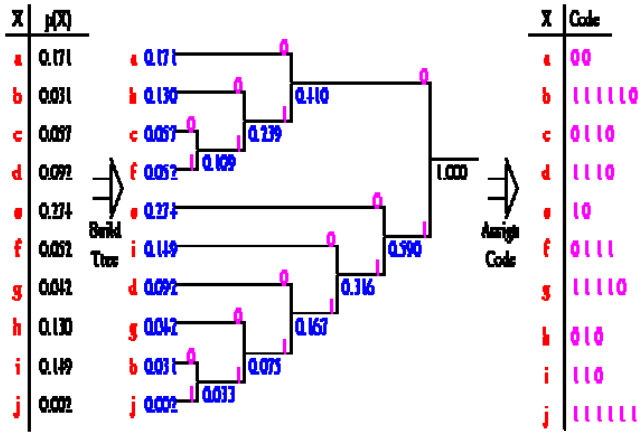
We are implementing Huffman coding using combination of three modules i.e. histogram,sorting and coder . First block i.e. histogram first find out the frequency of occurrence of the event.Sorting, then arranges these frequencies in ascending order. After sorting,finally,encoder generates the code based on the data given to it by the sorting block.In this way, we will generate the Huffman code.

II. HUFFMAN CODING

A Huffman encoder takes a block of input characters with fixed length and produces a block of output bits of variable length. It is a fixed-to-variable length code. Lempel-Ziv, on the other hand, is a variable-to-fixed length code. The design of the Huffman code is optimal (for a fixed blocklength) assuming that the source statistics are known a priori. The basic idea in Huffman coding is to assign short codewords to those input blocks with high probabilities and long codewords to those with low probabilities. A Huffman code is designed by merging together the two *least probable* characters, and repeating this process until there is only one character remaining. A code tree is thus generated and the Huffman code is obtained from the labeling of the code tree.

III. BINARY TREE

Final code tree is given below along with its probabilities and final code generation using binary tree method which is the basis of Huffman coding.



As shown above, letter j has the least probability. So,it has more number of bits to encode it. Also,e is having the highest probability. So,it has lower number of bits to encode it.

IV. BLOCK DIAG AND ITS EXPLANATION

Block diag which is used for huffman coding is shown below.

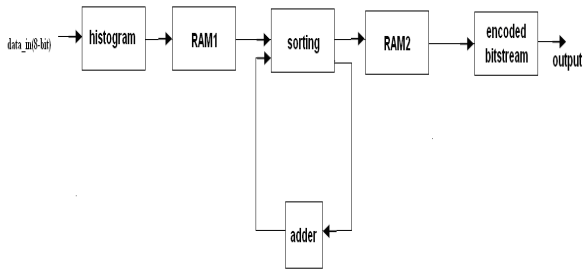


Fig 1) Block diag for huffman coding.

In this diag, input is 8 bit parallel data and output is serial data.

Histogram first takes 8 bit data as input and it finds the probability of ocurrence of different data.Output of histogram is given as input to the sorting ,which arranges the probabilities in ascending order and finally, sorted probabilities are given as input to the encoder,where,it finally generates the huffman code using a binary tree method.

Steps to build the Huffman tree:

1. Sort the frequencies into increasing order.

2. Choose the two smallest values.

3. Construct a binary tree with labeled edges.

4. Replace the two smallest values with their sum.

5. Getting a new sequence

6. Again take the two smallest values and construct a labeled binary tree.

7. Go to step 2 until remain no letter.

8. Finish!

In the fig. (a) sorting is done as shown and 1st step of tree is prepared

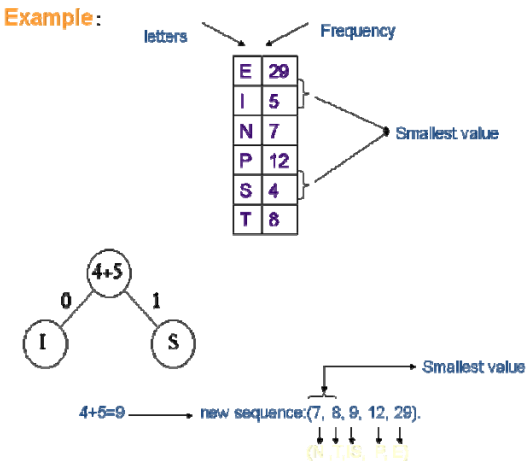


Fig. (a)

By obtaining the 1st step of tree again sorting and tree building is done as shown in fig.(b)

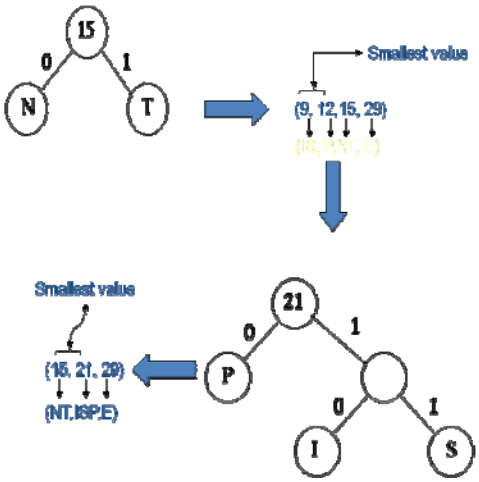


Fig. (b)

Again same procedure is repeated as shown in fig. (b) in fig. (c)

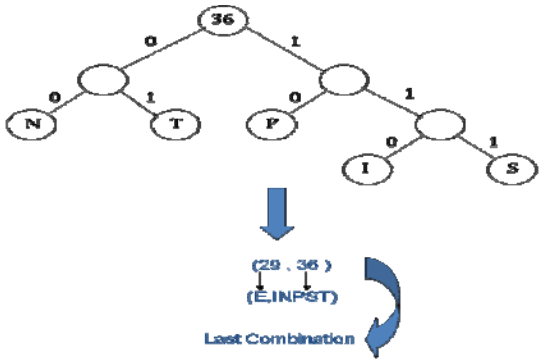
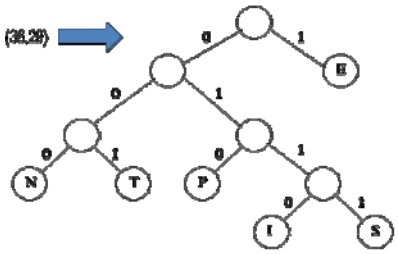


Fig.(c)

The final Huffamn tree is as shown in fig.



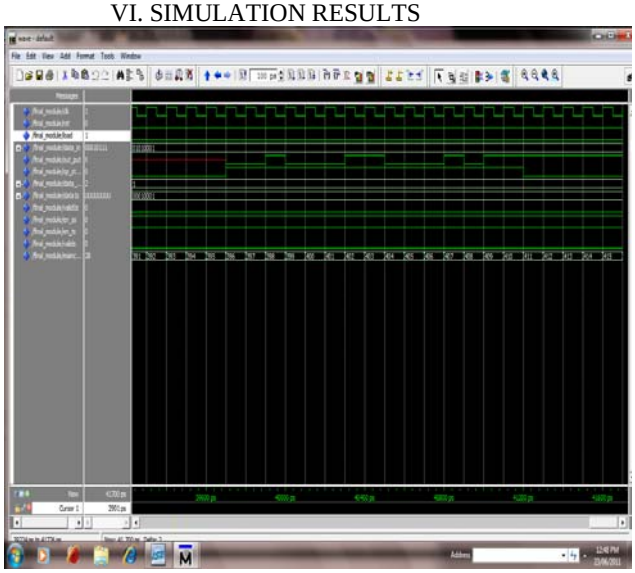
Result)

E	1
I	0110
P	010
N	000
S	0111
T	001

V. CALCULATIONS

We had 6 letters so we need 3 bits for each letter in normal coding. If the entire message is 65 character long so 3*65=195 bits to code but if we use “Huffman” the message require: 1*29+4*5+3*12+3*7+4*4+3*8=146 bits.
We have 100-(146/195)*100=25% of memory.

VI. SIMULATION RESULTS



Above fig. shows the simulation results if we take the input frequency as 5,10,6,15,17. Padding of zero is done for first three set of codewords.

VII. APPLICATIONS

1) It is used in mp3 and AA3 for sound and voice compression.
2) It is user in JPEG and MPEG for digital image compression.
3) Huffman compression is implemented in computer networks, modems and fax machines
4) Using Huffman coding in VLSI,fast video streaming on internet is possible.

REFERENCES

1) G.Wallace, The JPEG Still Picture compression Standard. April 1991, Vol.34 ,No.4 , Communication of the ACM.

2) CCITT Rec.T.81(1992 E), Digital compression and coding of Continuous-Tone Still images- Requirements and Guidelines(ISO/IEC 10918-1:1993(e),1992 pp. 18-21, 50-53, 88-93, 103-113.
3) E. Hashemian S. Sahni, S. Andersonfreed, “ Fundamentals of Data Structures in C” New York, 1993,pp.201.
4) T.Jeong, “ Implementation of low power adder design and analysis based on power reduction technique”,Microelectronics Journal,vol.39,pp.1880-1886, NOV. 2008.
5) K.K.Parhi-High-speed VLSI architectures for Huffman and viterbi decoders, IEEE Tran On circuits and systems II, vol 39/6 ,June 1999, pp 385-391.