Q11

a. Why would hardware compression be used in embedded systems?

In embedded systems, in particular instruction compression is important as instruction storage is expensive. We reduce the most commonly executed instructions to reduce energy in embedded systems.

b. What is the difference between lossy and lossless compression? Is Huffman coding lossless or lossy? Where else is Huffman coding used other than memory compressions?

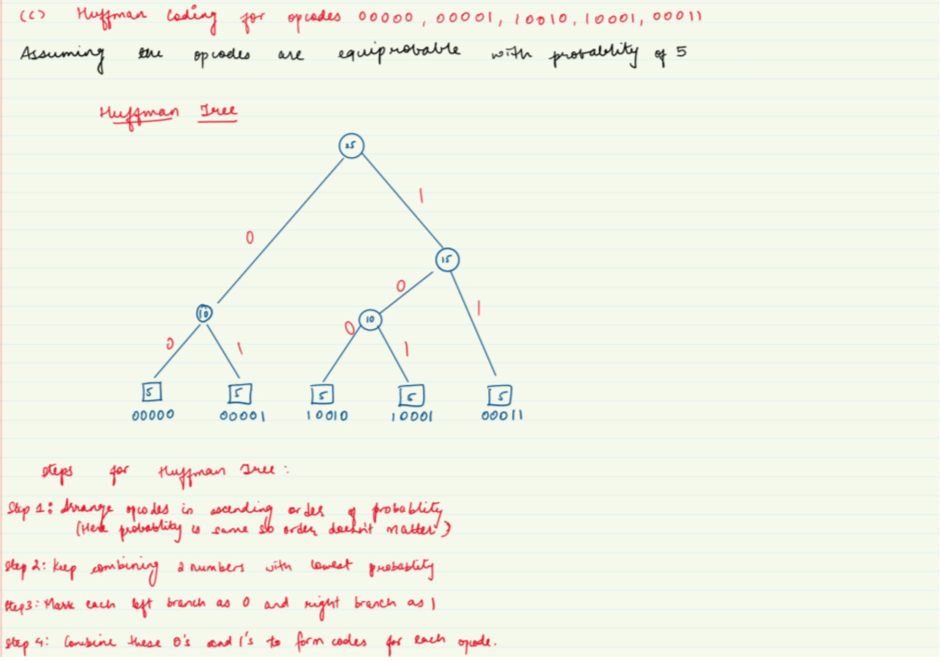
Lossy compression is based on a quality-of-service model, where it determines which data are critical to achieving a higher quality of service. Lossy compression greatly reduces the number of bits needed to represent data but cannot reproduce the same original quality as there is some information loss.

Lossless compression, on the other hand, requires that data be compressed and decompressed with no information loss compared to the originally compressed data.

Huffman coding is lossless as the data is recovered without any loss of information during decompression.

It is also used in Cryptography

c. For a given five-bit opcodes as 00000, 00001, 10010, 10001, 00011, use a Huffman code to encode them. Show the Huffman coding tree and the codes for each opcode. Assume that all opcodes are equally probable.



(c) Huffman coding for opcodes 00000, 00001, 10010, 10001, 00011

Assuming the opcodes are equiprobable with probability of 5

Huffman Tree

25
0
1
10
15
0
1
0
10
1
0
1
5
5
5
5
5
00000
00001
10010
10001
00011

steps for Huffman Tree :

Step 1: Arrange opcodes in ascending order of probablity
(Here probablity is same so order doesn't matter)

Step 2: Keep combining 2 numbers with lowest probablity

Step 3: Mark each left branch as 0 and right branch as 1

Step 4: Combine these 0's and 1's to form codes for each opcode.

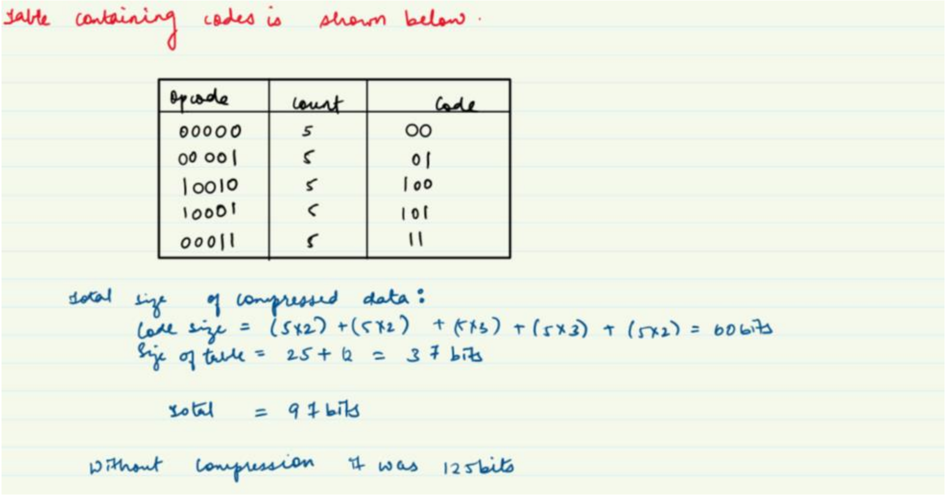


Table containing codes is shown below.

Opcode	Count	Code
00000	5	00
00001	5	01
10010	5	100
10001	5	101
00011	5	11

Total size of compressed data:

Code size = (5×2) + (5×2) + (5×3) + (5×3) + (5×2) = 60 bits

Size of table = 25 + 12 = 37 bits

Total = 97 bits

Without compression it was 125 bits