DCT IN VERILOG HDL

Abstract

DCT (Discrete Cosine Transform), a widely used lossy image compression technique, implemented in Verilog. The data retainment is approximately 99.95% for this implementation. A python script has been used to complement this process by converting an image into its BGR matrices format that is being fed into the verilog program. No processes related to DCT/IDCT has been implemented in the python script. The code is compatible with both grayscale and color images. We have also implemented decompression algorithm IDCT (Inverse DCT) in verilog. A python script is used to display and store the decompressed image.

Things to note

- The 8x8 matrix is represented using 4096 bit sized number, say X. Consider A to be flattened one dimensional array of 8x8 matrix. The first element of A is stored in first 64 bits of X, second element is stored in the next 64 bits and so on. (64 x 64 = 4096). So, Wherever we have used term matrix, It means we are using appropriate bit sized number.
- T is a constant floating point 8x8 matrix. Hence 1000*T is fed into verilog code. After computing TMT' ,we have divided it by 10^6.

Modules

1. mat_mul(c,a,b)

   Matrix multiplication of a and b

2. mat_transpose(b,a)

   Transpose for matrix a

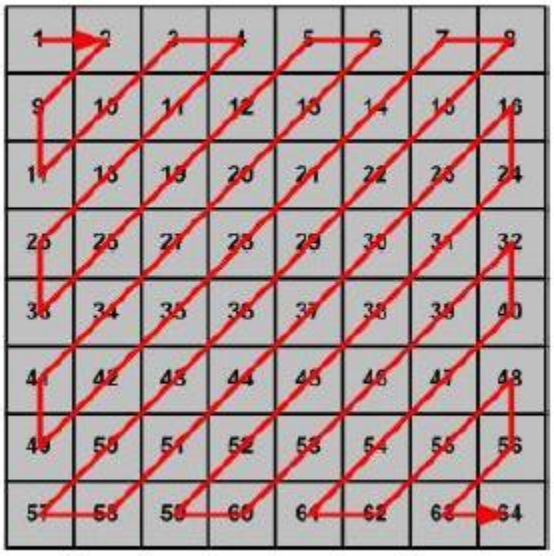
3. division(out,inp,q) and multiply(out,inp,q)

Element wise division/multiplication of inp and q. Rounding off has been implemented in division module.

4. encode(zig,W,X) and decode(X,C,d)

Encode returns the first 32 coefficients of W in a zig-zag pattern as shown in figure since most of the other coefficients are almost zero. X is the pattern. And Decode arranges the 32 dct coefficients back in their place and fills the remaining elements with zero. The same can be done with 16 coefficients for more compression

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64



5. dct_test1(C,M,T,T_,Q,x)

Main module controlling the operations to be done as per the algorithm. M,T,T_,Q,x are the inputs. C is the output. T , T_ , Q , x are constants.The dct_test1 takes 8x8 matrix M as major input and outputs the C which is the first 32 coefficients of the dct matrix. Input is 64 numbers and output is 32 numbers .Hence 50% compression.

6. idct_test1(N,C,T,T_,ss,Q)

This module is implemented as per the algorithm. C,T,T_,ss,Q are inputs and N is output. T , T_ , Q , x are constants.This module returns N, the decompressed image by decoding C. The decompressed image matrix will not be the same as the original matrix but

as an image it is the same one but with some loss.

7. dct_tb_final

   This Testbench module reads “Matrix.txt” which contains constants, ”image.txt” which contains image matrix. And also outputs “out_dct.txt” which contains compressed image.

8. idct_tb_final

   This Testbench module reads “Matrix.txt” which contains constants, “out_dct.txt” which contains compressed image. And also outputs “out_idct.txt” which contains decompressed image.

References

- https://www.math.cuhk.edu.hk/~lmlui/dct.pdf
- https://www.geeksforgeeks.org/discrete-cosine-transform-algorithm-program/
- PPT for reference: https://docs.google.com/presentation/d/1BwLgxYBSefvEc3VN8k9dOJzzW3v2QyhgxcZhMWfgLJs/edit#slide=id.p