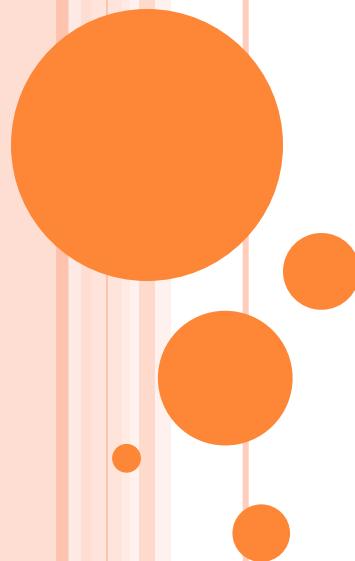
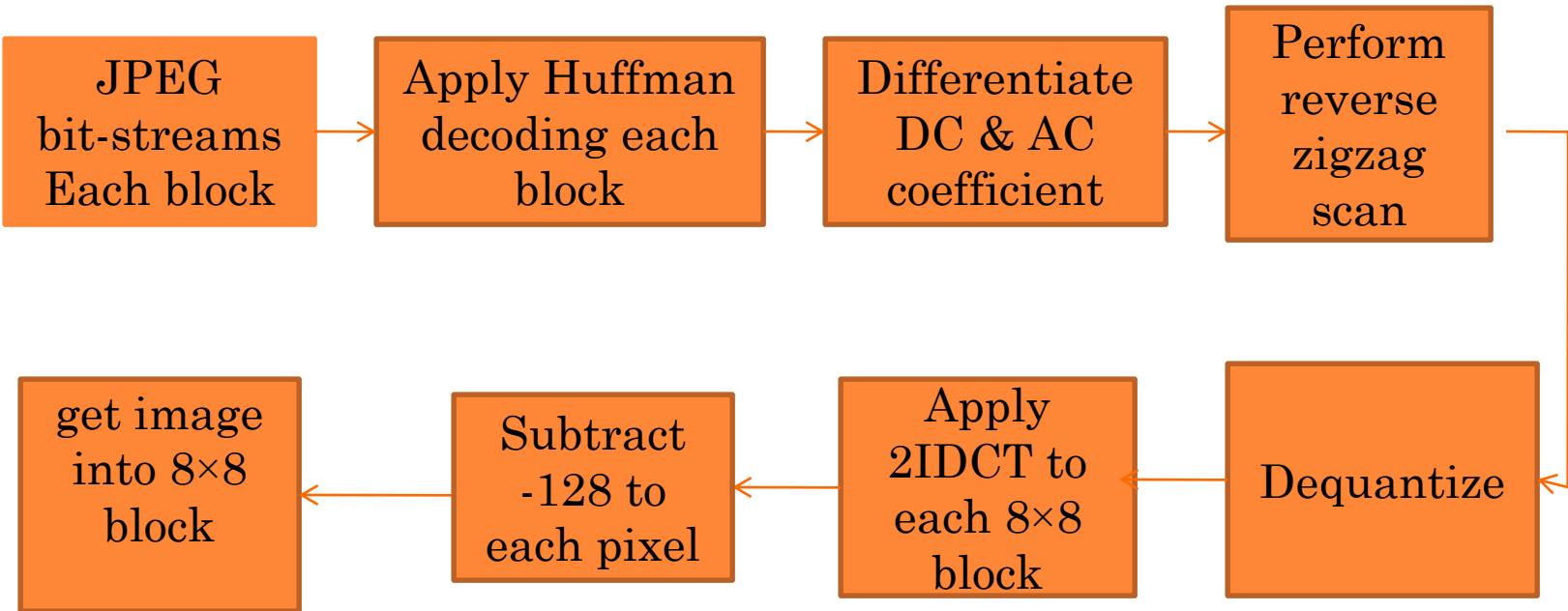


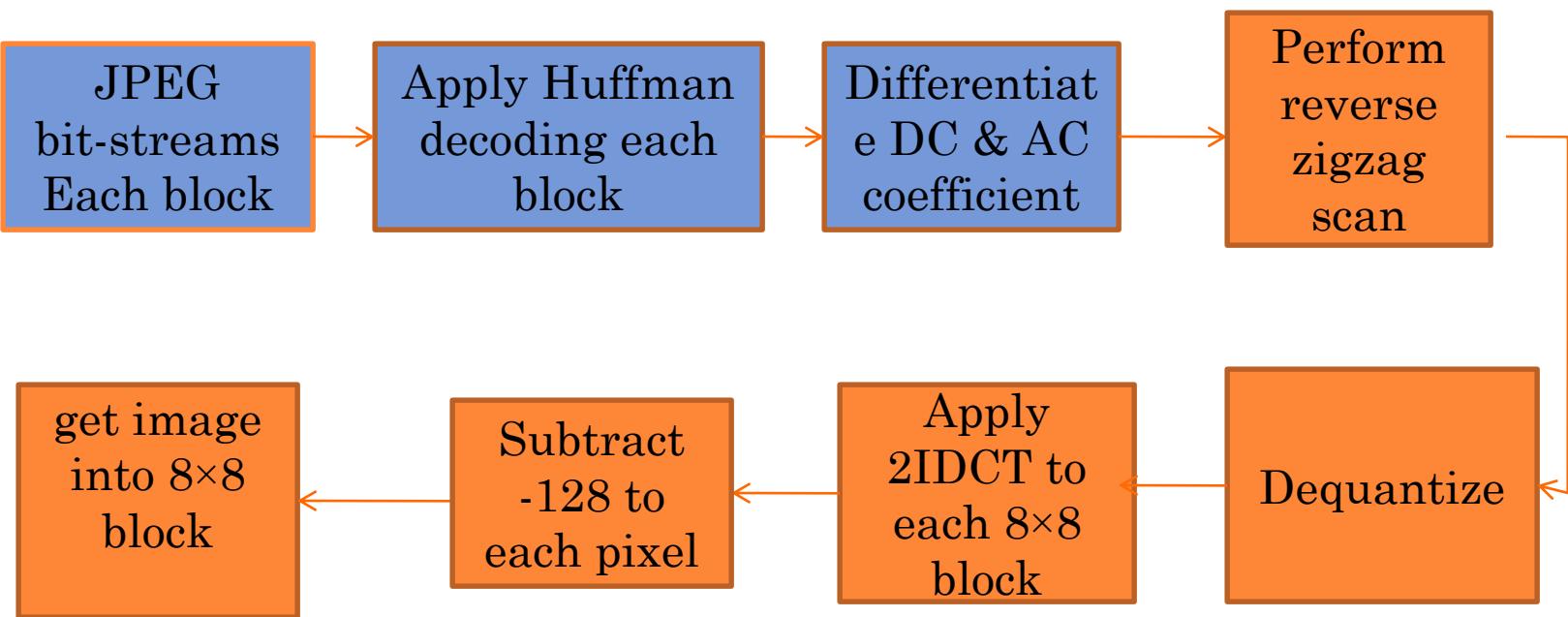
# The JPEG Decoding Mechanism



# THE BASELINE JPEG ALGORITHMIC DECODING FLOW-CHART



# Specific Example



## Cont....

Let us suppose we have the following array of bits encoded in the disk.

```
1010110 0100 001 0100 0101 100001 0110 100011 001 100011 001  
001 100101 11100110 110110 0110 11110100 000 1010
```

Get the block following

**-26,-3,1,-3,-2,-6,2,-4,1,-4,1,1,5,0,2,0,0,-1,2,0,0,0,0,-1,-1,EOB**

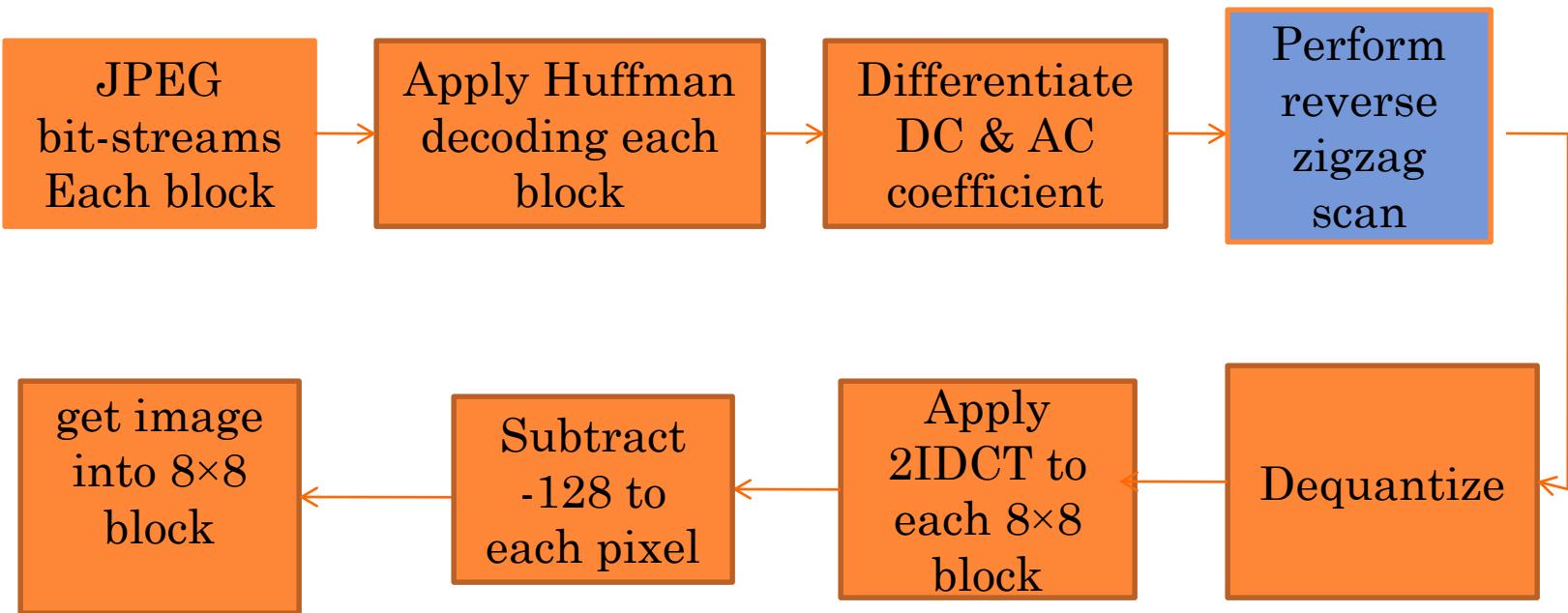
→ DC Coefficient

→ AC Coefficient

→ End of Block



# Cont.....



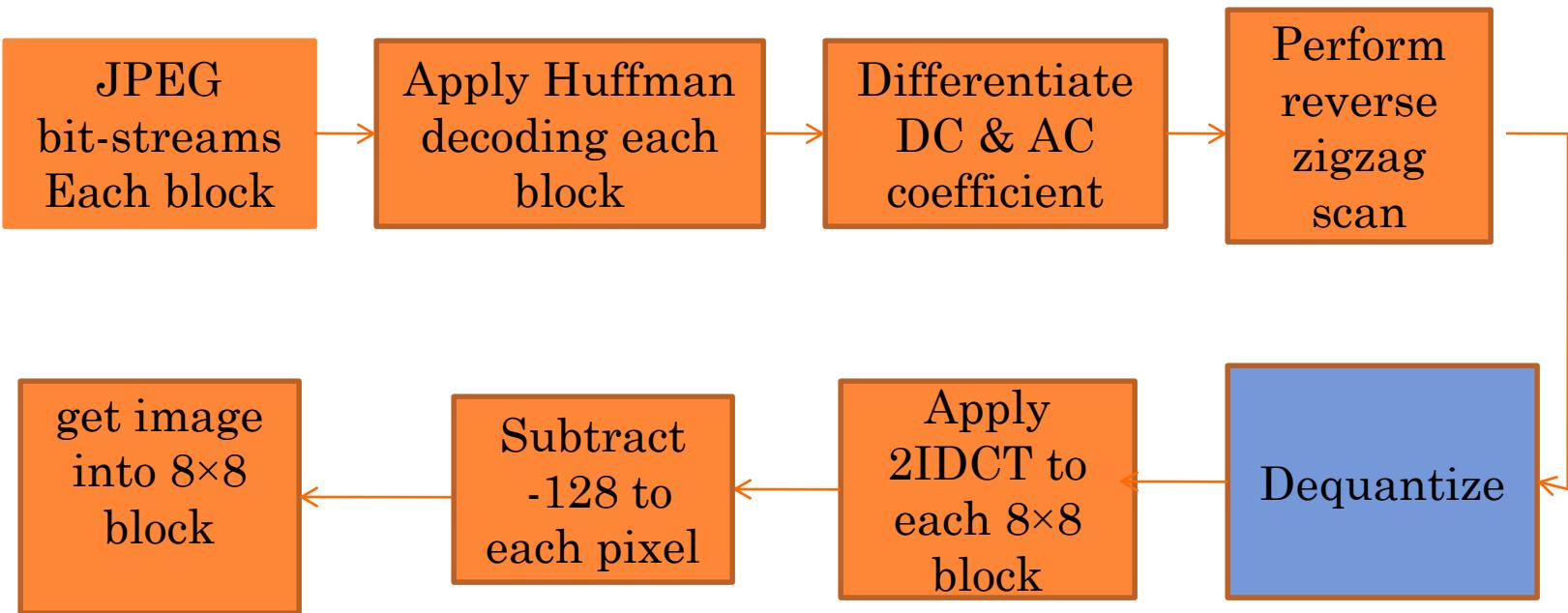
## Cont.....

In this step, reverse zigzag ordering of the dequantized block should be performed.

$$B = \begin{matrix} -26 & -3 & -6 & 2 & 2 & 0 & 0 & 0 \\ 1 & -2 & -4 & 0 & 0 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -4 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$$



## Cont.....

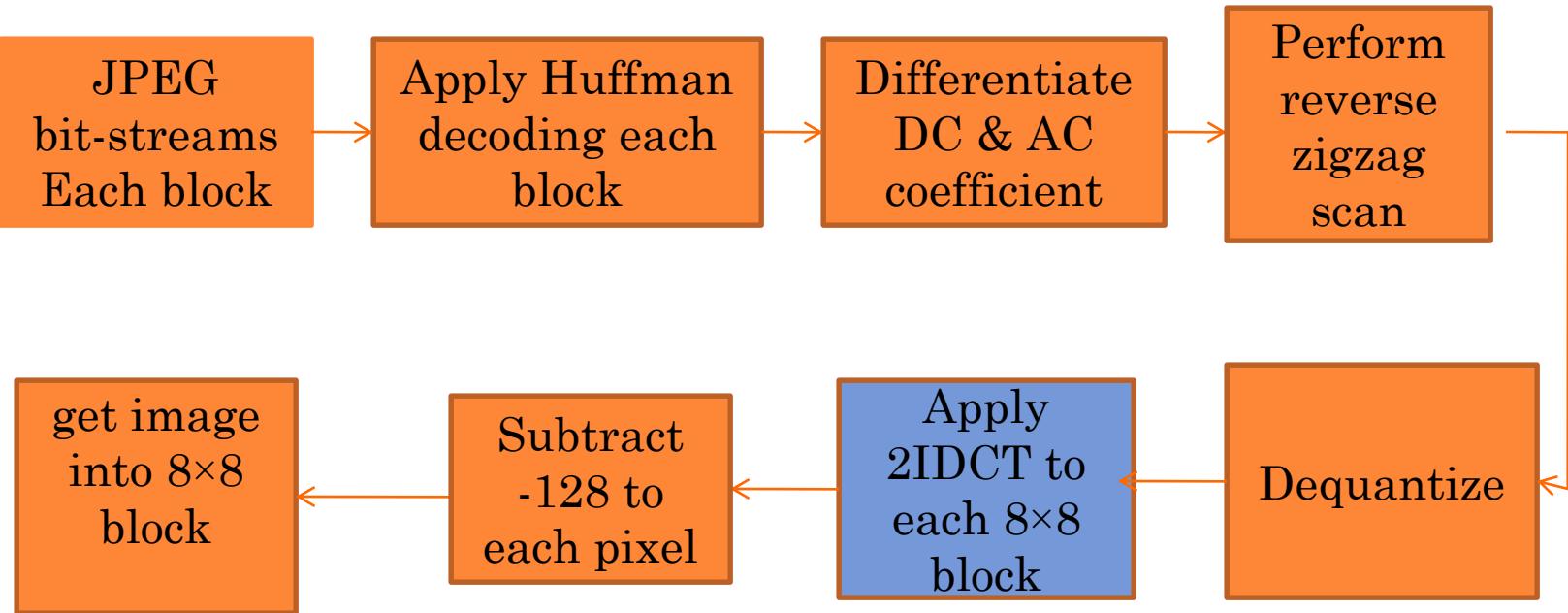


# Cont.....

$$B = \begin{bmatrix} -26 & -3 & -6 & 2 & 2 & 0 & 0 & 0 \\ 1 & -2 & -4 & 0 & 0 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -4 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} -416 & -33 & -60 & 32 & 48 & 0 & 0 & 0 \\ 12 & -24 & -56 & 0 & 0 & 0 & 0 & 0 \\ -42 & 13 & 80 & -24 & -40 & 0 & 0 & 0 \\ -56 & 17 & 44 & -29 & 0 & 0 & 0 & 0 \\ 18 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$Q = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}.$$

# Cont.....



## Cont.....

Inverse Discrete Cosine Transform should now be applied to each block as follows-

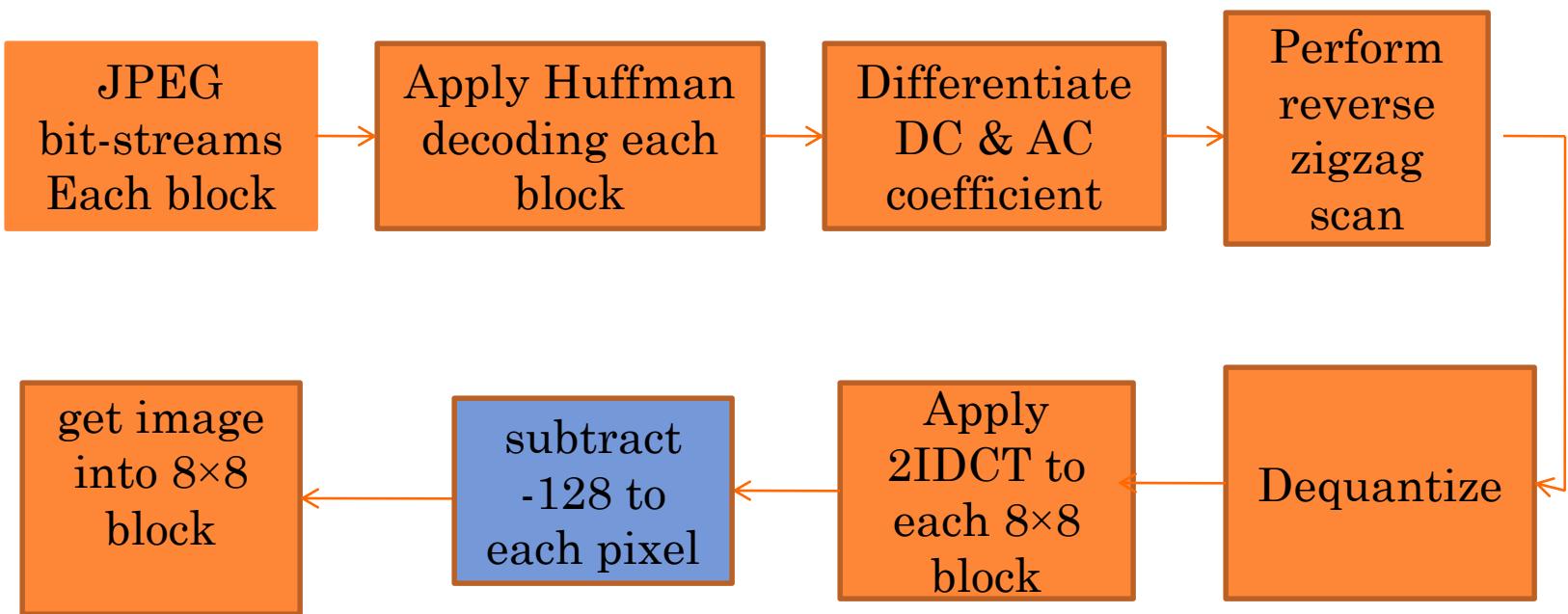
$$S_{xy} = \frac{1}{4} \sum_{x=0}^7 \sum_{y=0}^7 C_u C_v S_{vu} \cos \frac{(2x+1)u\pi}{16} \cos \frac{(2y+1)v\pi}{8}$$

Where,  $C_u, C_v = \frac{1}{\sqrt{2}}$  for  $u, v = 0$ ; otherwise  $C_u, C_v = 1$ .

-416	-33	-60	32	48	0	0	0	-70	-64	-61	-64	-69	-66	-58	-50
12	-24	-56	0	0	0	0	0	-72	-73	-61	-39	-30	-40	-54	-59
-42	13	80	-24	-40	0	0	0	-68	-78	-58	-9	13	-12	-48	-64
-56	17	44	-29	0	0	0	0	-59	-77	-57	0	22	-13	-51	-60
18	0	0	0	0	0	0	0	-54	-75	-64	-23	-13	-44	-63	-56
0	0	0	0	0	0	0	0	-52	-71	-72	-54	-54	-71	-71	-54
0	0	0	0	0	0	0	0	-45	-59	-70	-68	-67	-67	-61	-50
0	0	0	0	0	0	0	0	-35	-47	-61	-66	-60	-48	-44	-44



# Cont.....



# Cont.....

We subtract **-128** to each pixel in this step.

-70	-64	-61	-64	-69	-66	-58	-50	58	64	67	64	59	62	70	78
-72	-73	-61	-39	-30	-40	-54	-59	56	55	67	89	98	88	74	69
-68	-78	-58	-9	13	-12	-48	-64	60	50	70	119	141	116	80	64
-59	-77	-57	0	22	-13	-51	-60	69	51	71	128	149	115	77	68
-54	-75	-64	-23	-13	-44	-63	-56	74	53	64	105	115	84	65	72
-52	-71	-72	-54	-54	-71	-71	-54	76	57	56	74	75	57	57	74
-45	-59	-70	-68	-67	-67	-61	-50	83	69	59	60	61	61	67	78
-35	-47	-61	-66	-60	-48	-44	-44	93	81	67	62	69	80	84	84



# Conclusion

Any difference between the original and reconstructed sub image are a result of the lossy nature of the JPEG compression and decompression process .In this example, the error range from -14 to +11 and are distributed follow

-6	-9	-6	2	11	-1	-6	-5
7	4	-1	1	11	-3	-5	3
2	9	-2	-6	-3	-12	-14	9
-6	7	0	-4	-5	-9	-7	1
-7	8	4	-1	11	-4	3	-2
3	8	4	-4	2	11	1	1
2	2	5	-1	-6	0	-2	5
-6	-2	2	6	-4	-4	-6	10



Thanks to all

