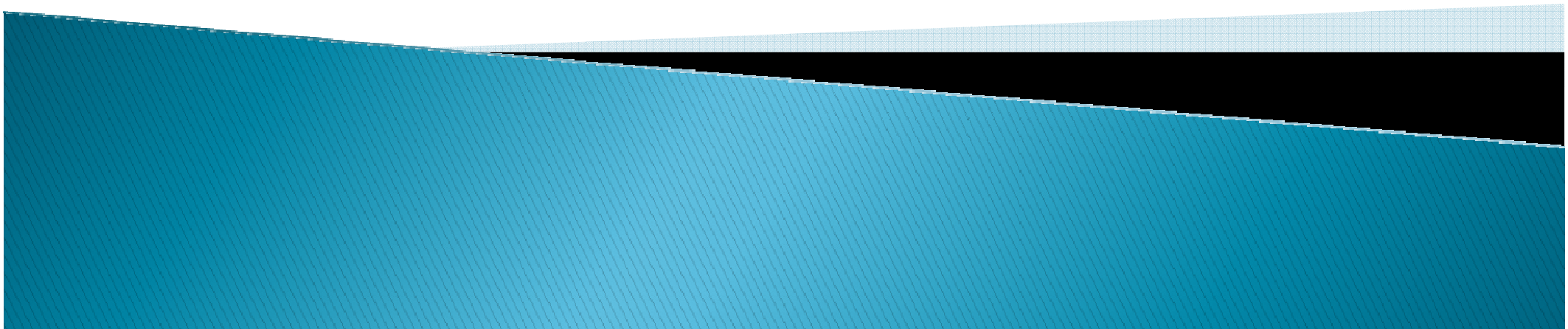
DATA-PATH AND MEMORY ERROR COMPENSATION TECHNIQUE FOR LOW POWER JPEG IMPLEMENTATION

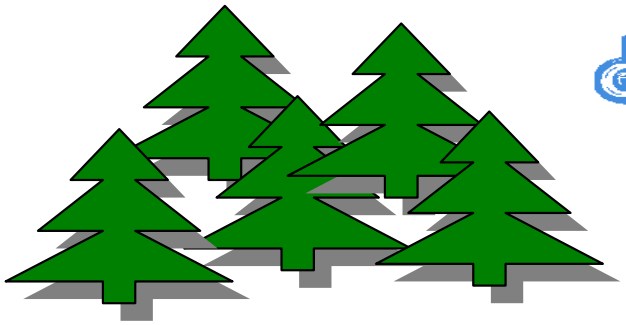
Yunus Emre
Chaitali Chakrabarti
Arizona State University
School of Electrical Computer and Energy Engineering

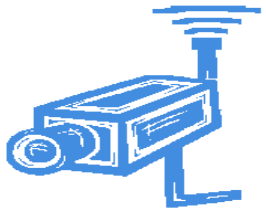


Low Energy Codecs

Portable Devices

Surveillance





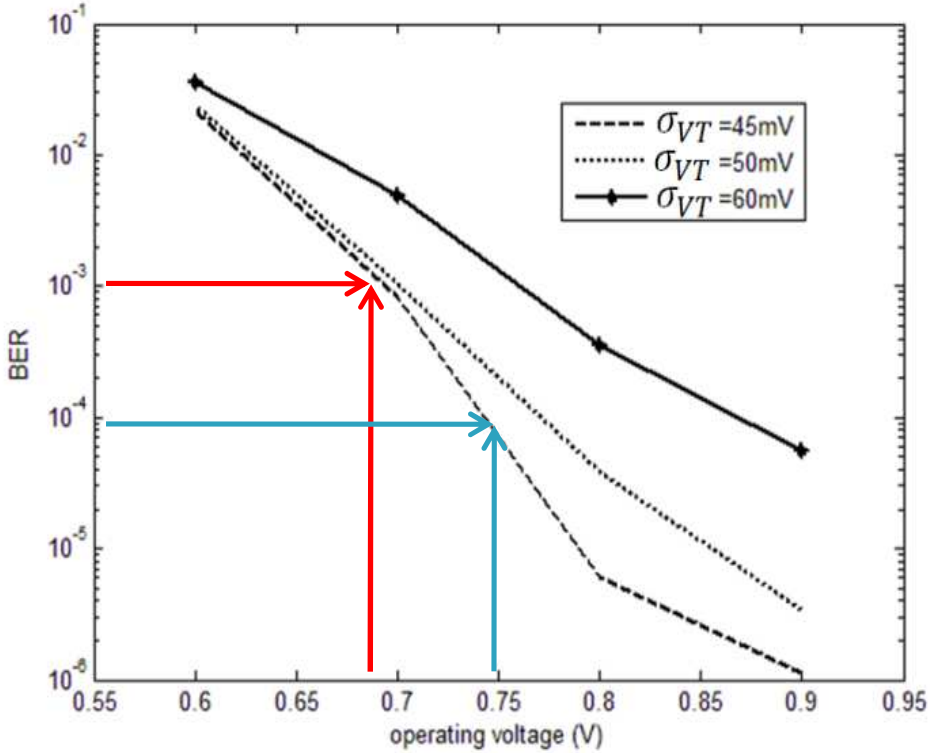
Low Energy
Low Quality

Medium Energy
Medium Quality

High Energy
High Quality

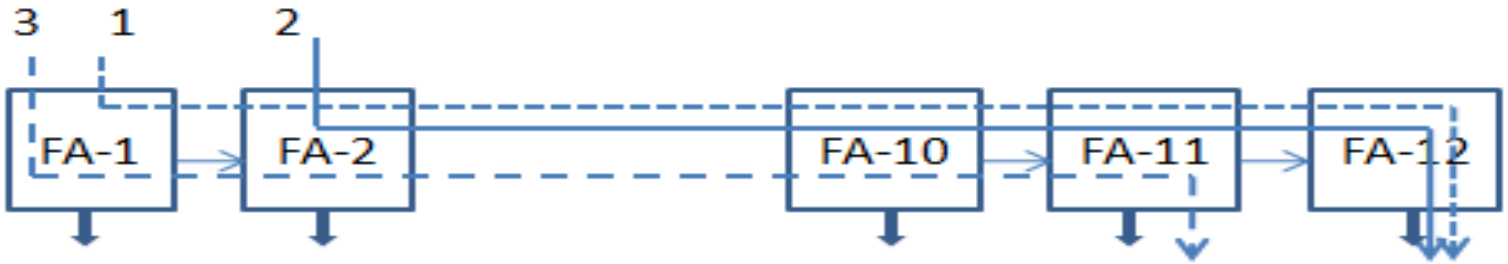
Source of Errors – Memory Failure

- Memory failure rate is affected by several factors
  - process variations
    - random dopant fluctuation (RDF),
    - length, width and oxide thickness
- Threshold voltage variation increases with technology and voltage scaling



Failure rate of a 32nm, 6T SRAM Cell for different voltage levels

Data-Path Failure Model



12-bit adder used in DCT

t_{FA} : nominal delay of a full adder t_{SYS}: delay due to systematic variation

$t_{r,-}$: delay due to random variation σ_{FA} variance of $t_{r,-}$

Delay of each carry chain starting from x FA and ending at y FA:

$$T_{chain}(x,y) = (x-y) * (t_{FA} + t_{SYS}) + (t_{r,x} + \cdots + t_{r,y})$$

$$T_{chain}(\Delta) = \Delta * (t_{FA} + t_{SYS}) + \sqrt{\Delta} \times t_r \quad \text{where} \quad \Delta = x - y$$

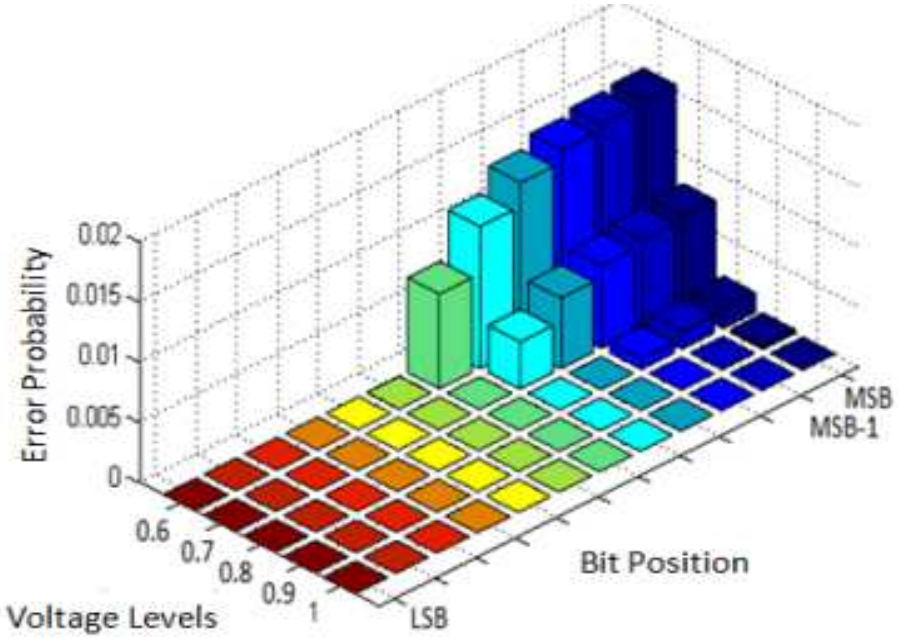
Thus $T_{chain}(\Delta)$ is a Gaussian variable with $\mu = \Delta * (t_{FA} + t_{SYS})$ and $\sigma = \sqrt{\Delta} \times \sigma_{FA}$ Delay of any chain is represented by 12 different distributions $T_{chain}(1)$ to $T_{chain}(12)$

Data-Path Failure Model

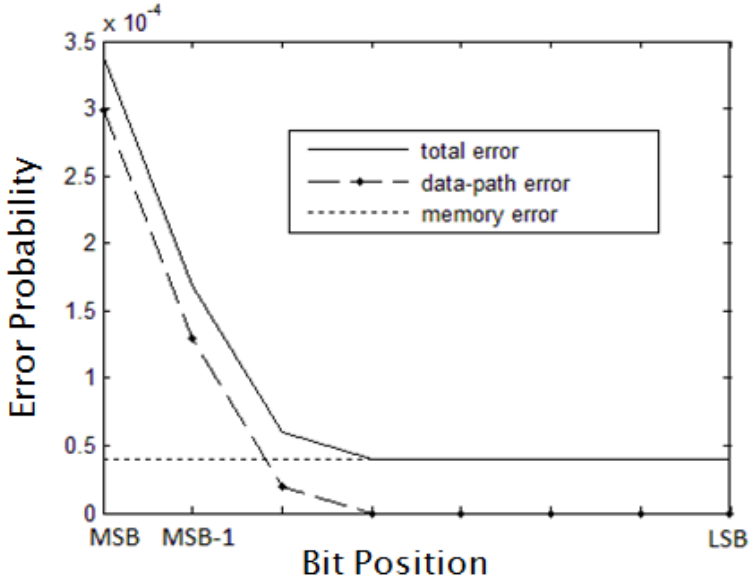
- The probability of error for bit k using Bayes' theorem and summing over all paths

$$p(t_k > t_{crt}) = \sum_{z=1}^{k} p(T_{chain}(z) > t_{crt} | chain = z) \times p(chain = z)$$

where t_k :path delay of the k bit and $p(chain = z) = \frac{1}{2^z}$



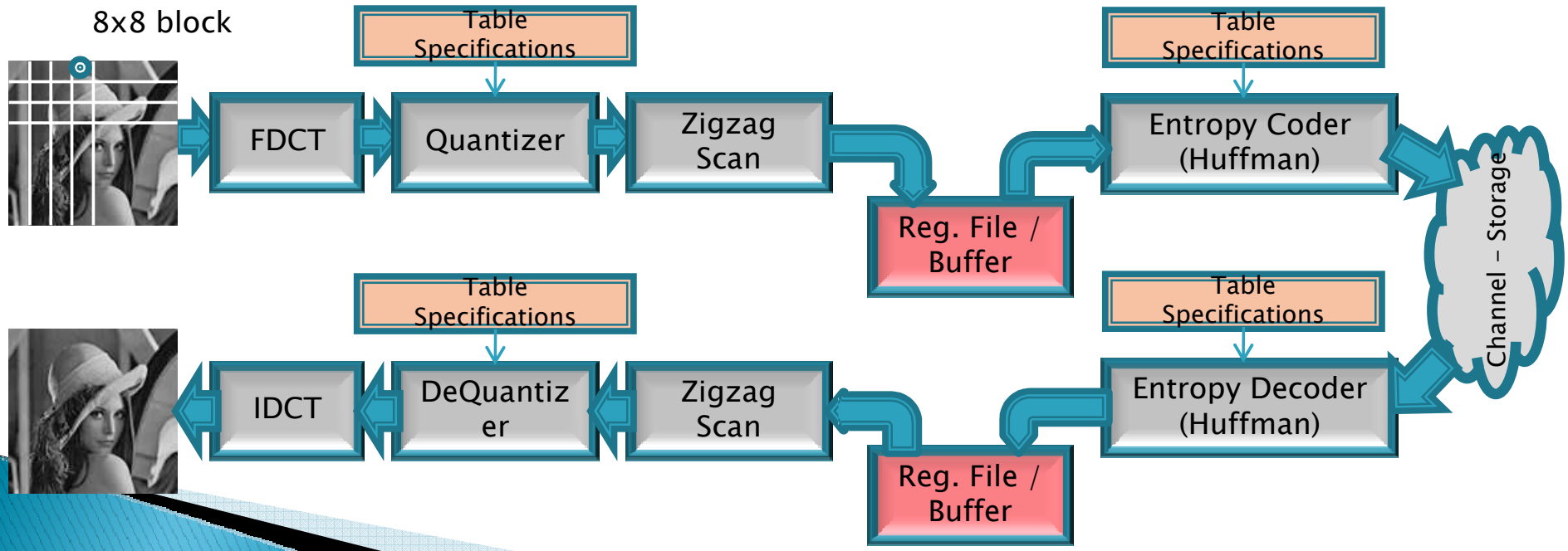
Probability of error distribution for 12-bit RCA for t_{crt} of 400ps



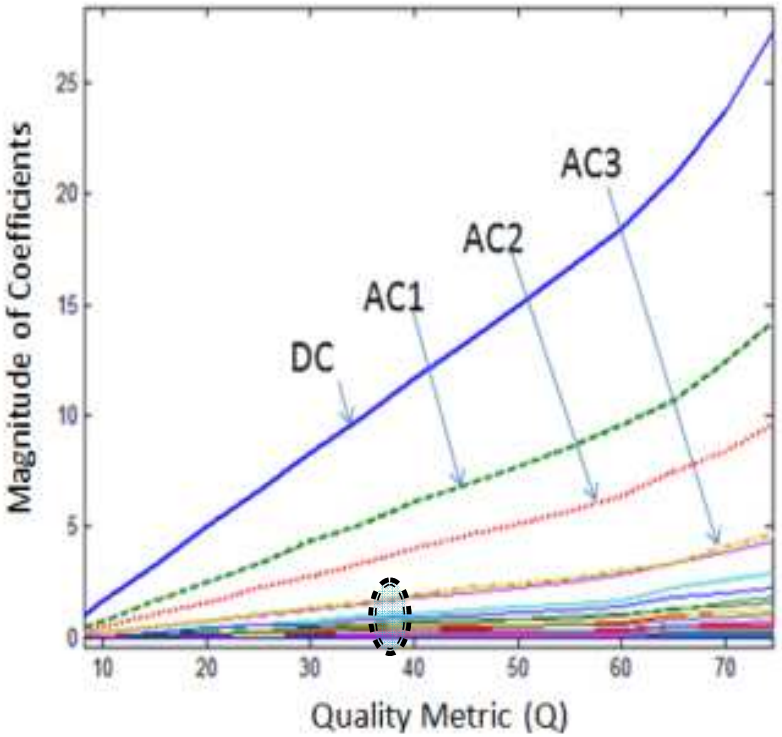
Pdf of errors from MSB to LSB for BER $= 10^{-4}$

Baseline JPEG

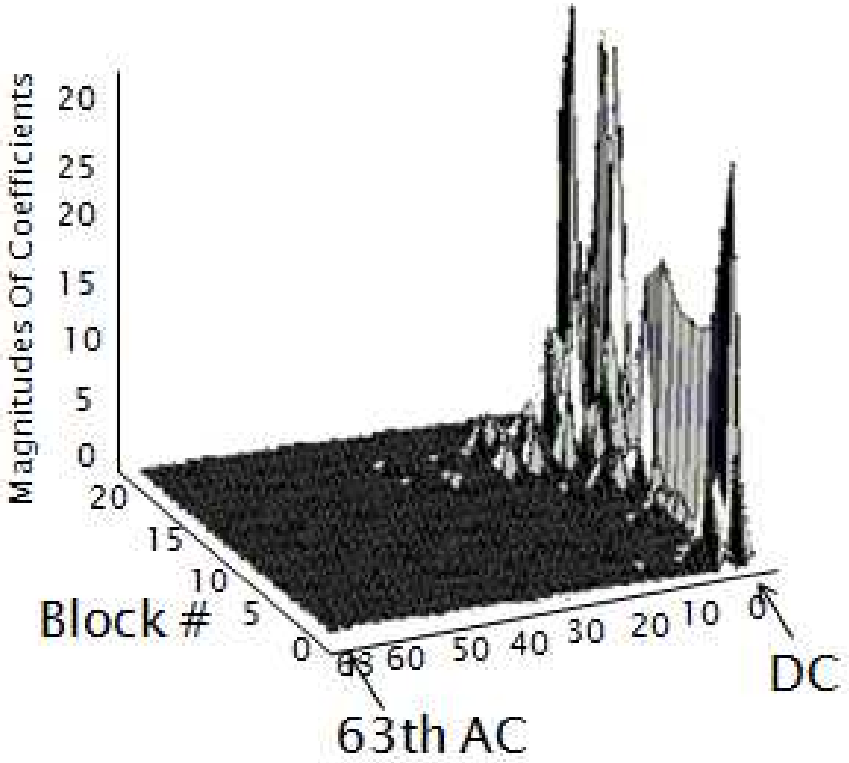
- Main Kernels:
  - Discrete Cosine Transform (DCT)
  - De/Quantizer
  - Zig-Zag Scan
  - Entropy Coder: Huffman Coding



Compensating for Errors



Two adjacent AC coefficients after zig-zag scan have similar values for wide range of Q



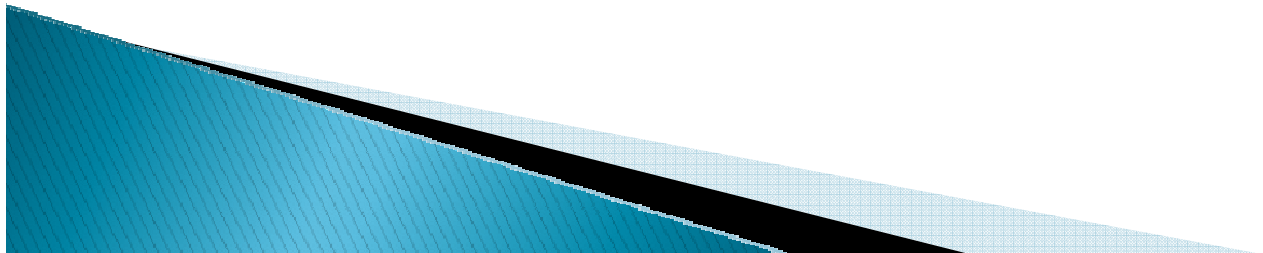
AC coefficients representing same frequencies at neighboring blocks usually have similar values

Algorithm-Specific Technique: Step 1

- Number of sign extension bits is determined by the quantization step

Quantizer	Group-1	Group-2	Group-3	Group-4
Q=<5	5	4	3	2
5<Q=<15	6	5	4	3
15<Q=<30	7	6	5	4
30<Q=<55	8	7	6	5
55<Q=<70	8	7	6	6

- Detect bit errors that occur in the sign extension bits of coefficients
- When k bit representation is sufficient for Group l, then by definition, the sign extension bits k to MSB should be all zero for a positive number and all one for a negative number.
- Step 1: Pick three bits from the sign extension bits to find the correct sign extension bits



Algorithm-Specific Technique: Step 2

Coefficients that are adjacent to each other have similar magnitudes

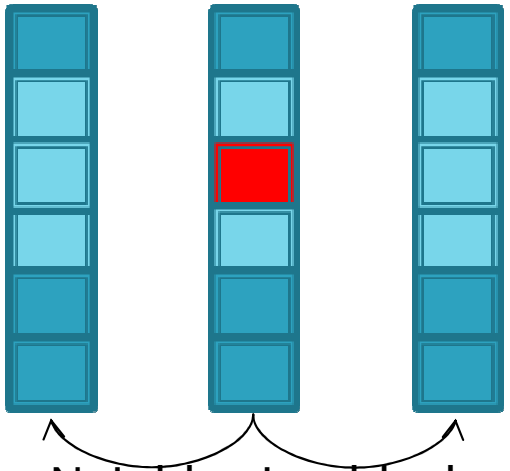
- Detect an error when there is an abnormal increase in magnitude in one of the coefficients
- Use the neighboring DCT values to correct

```
BEGIN
1.   Initialize Parameters: Thresholds (THR_i) i=1,2,3,4
2.   FOR each AC coefficient LOOP
3.     IF ( AC(k,j)-( AC(k,j+1) + AC(k,j-1) )/2 > THR_i )
4.       IF ( AC(k,j)-( AC(k+1,j) + AC(k-1,j) )/2 > THR_i )
5.         AC(k,j)=( AC(k,j+1) + AC(k,j-1) )/2
END
```

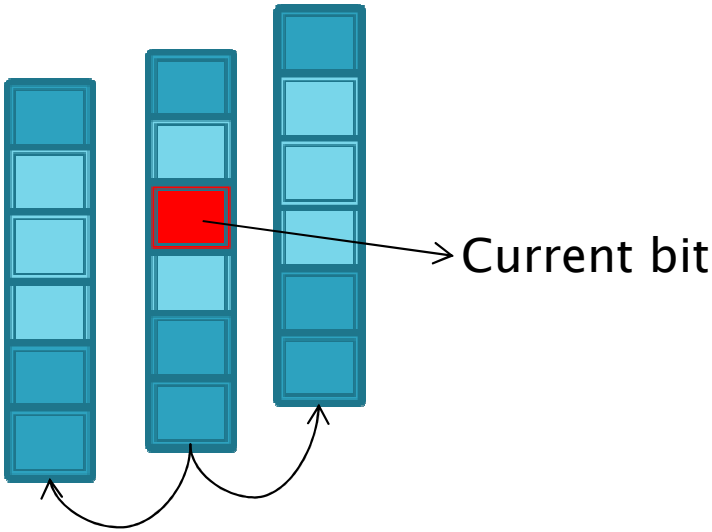
Algorithm-Specific Technique: Step 3

Used to combat errors due to memory failures in lower order bits

- Check neighbors of the current bit; 3 bit from 2 neighboring blocks same frequency, 3 bits from same block but 2 neighboring frequencies and 2 adjacent bits from the same coefficient.
- When another non-zero bit is detected in the search space, it decides that the current bit is not in error except when the non-zero bit is one of a group of three consecutive 1s.



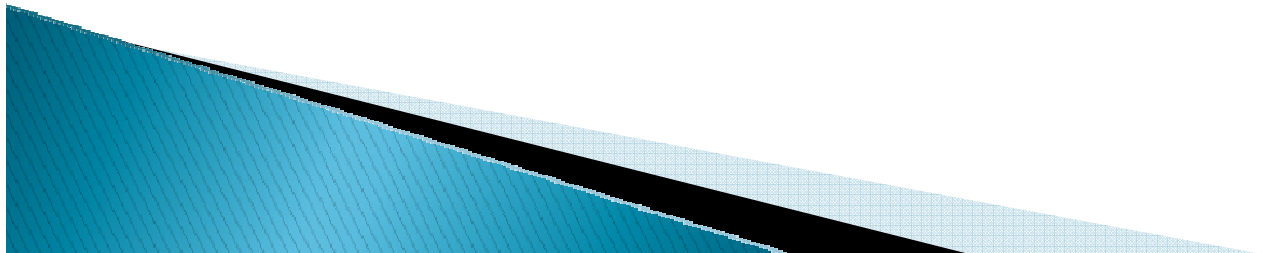
Neighboring blocks
Same frequency



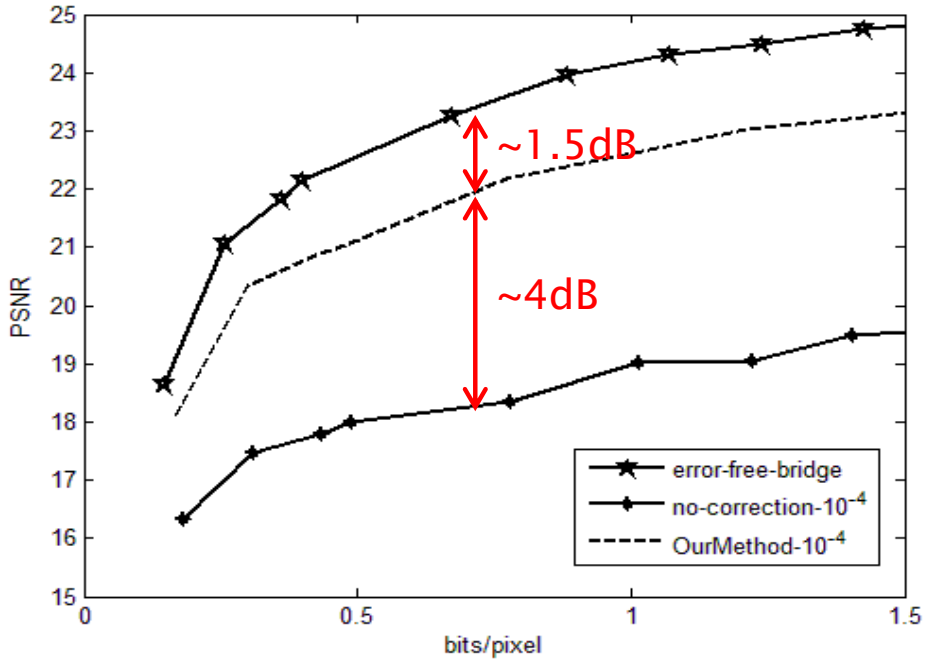
Same blocks
Neighboring frequency

Simulation Setup

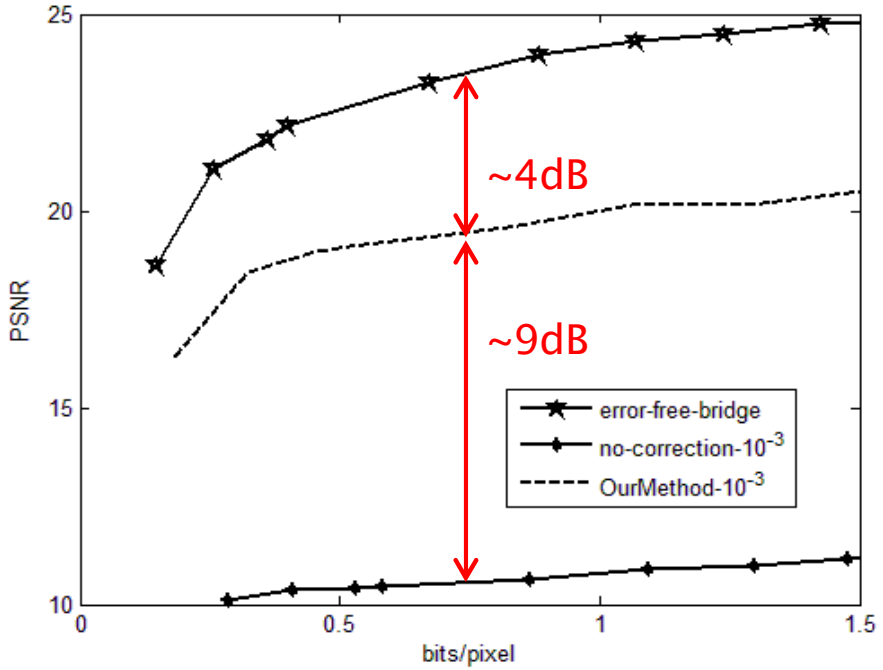
- Quality Performance
  - Baseline JPEG
  - PSNR vs Compression Rate
  - Bridge, Baboon, Lena and Pepper Images
- Hardware Performance
  - Design Compiler from Synopsys
  - 45nm HP Nangate library



Simulation Results-1



Bridge image: Performance for BER 10^{-4}



Bridge image: Performance for BER 10^{-3}

Algorithm-specific techniques improve the PSNR quality by 4dB compared to the no correction case

Algorithmic-specific techniques improve the PSNR quality by 9dB at 0.75bpp compared to the no-correction case

Simulation Results-2

PSNR VALUES OF DIFFERENT TECHNIQUES AT 0.75 BPP COMPRESSION RATE FOR BER=10^{-4}

	Error-Free	No-Correction	Proposed Technique
Bridge	23.5	18.2	22.1
Baboon	23.7	18.6	22.4
Lena	34.5	27.1	32.3
Pepper	32.5	25.7	30.4

Algorithm-specific technique achieves 4.4 dB improvement on average

Circuitry Overhead

	Majority Voter	Coefficient Comparator	Magnitude Correction Unit
Area (um^2)	24	767	104
Worst-case delay (ps)	42	459	27
Active Power (uW)	3.6	96	6.5
Leakage Power (uW)	0.1	2.7	0.4

Conclusion

- The errors in DCT and memory units have been characterized and a pdf of the errors is derived as a function of the bit position.
- The proposed algorithm-specific technique exploits the characteristics of the quantized coefficients including similarity between neighboring coefficients in the same block and across blocks and the fact that high frequency coefficients have smaller values.
- The technique improves PSNR performance by approximately 4dB when BER=10^{-4}, compared to the no-correction case with a degradation of less than 2dB in PSNR compared to the error-free case.

THANK YOU

Questions?