

THIẾT LẬP KHỐI LƯỢNG TỬ TRONG CHUẨN NÉN ẢNH JPEG2000 TRÊN BỘ MẠCH FPGA IMPLEMENT HARDWARE THE QUANTIZATION OF THE JPEG2000 ENCODER ON FPGA

Bạch Tuấn Đồng^{1,2}, Nguyễn Minh Khánh Ngọc², Nguyễn Bá Tuyên¹, Lê Mỹ Hà¹

¹Trường đại học Sư phạm kỹ thuật TP. Hồ Chí Minh

²Trung tâm Nghiên cứu và Đào tạo Thiết kế Vi mạch- ĐHQG TP. Hồ Chí Minh

Ngày tòa soạn nhận bài 24/8/2015, ngày phản biện đánh giá 20/9/2015, ngày chấp nhận đăng 08/10/2015

TÓM TẮT

Lượng tử là một phần quan trọng cần thiết trong chuẩn nén ảnh JPEG2000. Quá trình lượng tử làm giảm số bit cần biểu diễn trong tất cả các sub-band. Bài báo này đã tìm 2 giá trị mantissa và exponent để cho kết quả nén hình ảnh tốt nhất cho từng mức biến đổi FDWT trong chuẩn nén ảnh JPGE2000. Mặt khác trong bài báo hỗ trợ chia hình ảnh ra dưới dạng code-block có kích thước 16x16, 32x32, 64x64. Quá trình chia code-block hỗ trợ thực hiện mã hóa tier 2 hiệu quả trong chế độ thực hiện thời gian thực. Thực nghiệm hóa hệ thống trên bảng mạch FPGA là một bước cần thiết để minh chứng cho hiệu quả của phương pháp đề xuất. Kiểm tra thiết kế sử dụng matlab, Modelsim VCS và Quartus II.

Từ khóa: Lượng tử hóa, FDWT, JPEG2000, bước nhảy, sub-band, ROI.

ABSTRACT

In the JPEG2000 standard for image compression, quantization is a necessary lossy technique to significantly reduce bit number in all sub-bands. In this paper, mantissa and exponent coefficients have been found for the best solution for the 8-bit and 16-bit images in the JPEG2000 image compression standard. Furthermore, the method also divides image data into code-blocks sizing 16x16, 32x32, and 64x64. The code-block process supports the real time tier-2 encoding implementation. Experiment has been done on the FPGA to demonstrate the effectiveness of this proposed method. Matlab, Modelsim VCS tools and Quartus II tools have also been used for testing.

Keywords: Quantization, FDWT, JPEG2000, step size, sub-band, ROI.

1. GIỚI THIỆU LƯỢNG TỬ HÓA TRONG CHUẨN JPEG2000

Sau quá trình biến đổi wavelet rời rạc, tất cả các sub-band được lượng tử trong chế độ nén tổn hao nhằm làm giảm độ chính xác của các sub-band, qua đó giảm được số bit cần biểu diễn. Lượng tử hóa các sub-band DWT là một trong những nguồn gây tổn hao trong hệ thống nén ảnh JPEG2000. Nén mất dữ liệu (lossy) sẽ cho tỷ lệ nén cao nhưng đồng thời cũng làm giảm chất lượng của ảnh được phục hồi bởi lượng thông tin đã bị mất. nén không mất dữ liệu (lossless) trong quá trình

nén không tổn hao hình ảnh sau khi phục hồi có PSNR (peak signal-to-noise ratio) là vô cùng. Đây là một điểm mạnh của nén ảnh theo chuẩn JPEG2000 so với JPEG

Công thức biến đổi bước nhảy lượng tử [1]

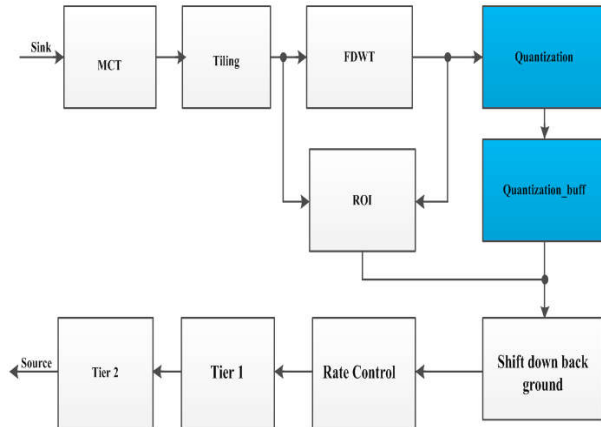
$$\Delta = 2^{R_b - \varepsilon_b} \left(1 + \frac{\mu_b}{2^{11}}\right) \quad (1)$$

R_b Độ sâu của bit ảnh.

μ_b Có giá trị được biểu diễn 11bits.

ϵ_b Có giá trị được biểu diễn 5 bits

Sau khi dữ liệu được lượng tử thì tiến hành chia thành từng code-block nhỏ. Trong thiết kế hỗ trợ code-block 16x16, 32x32 và 64x64 pixel.



Hình 1. Sơ đồ khối hệ thống JPEG2000

2. GIẢI PHÁP THIẾT KẾ PHẦN CỨNG

2.1 Thiết kế khối lượng tử

Công thức tính độ sâu của ảnh [1]

$$R_b = R_1 + \log_2(\text{gain}_b) \quad (2)$$

Trong đó giá trị $\log_2(\text{gain}_b)$

Có giá trị trong Bảng 1

Bảng 1. Giá trị gain_b

Sub-band	gain_b	$\log_2(\text{gain}_b)$
Level LL	1	0
Level LH	2	1
Level HL	2	1
Level HH	4	2

Trong bảng trên thì ta thấy giá trị $\log_2(\text{gain}_b)$ của sub-band chỉ thay đổi trong một level. Trong các level khác nhau thì quy luật này vẫn đúng theo Bảng 1. Giá trị trong Bảng 2 là giá trị R_b, R_T

Bảng 2. Giá trị R_1

Sub-band	R_1	$\log_2(\text{gain}_b)$	R_1
LL	8	0	8
LH	8	1	9
HL	8	1	9
HH	8	2	10
LL	16	0	16
LH	16	1	17
HL	16	1	17
HH	16	2	18

Trong Bảng 3 cung cấp những giá trị mantissa và exponent cần thiết để thực hiện phép toán trong công thức (1). Trong đó mantissa là 5 bits nên có 2^5 giá trị lựa chọn và exponent là 11 bits nên có 2^{11} giá trị lựa chọn, việc lấy 2 giá trị này cho phù hợp từng level là một điều khó khăn, sau nhiều quá trình thực nghiệm bài báo đã tìm ra được những giá trị mantissa và exponent phù hợp cho từng level được thể hiện trong Bảng 3. Giá trị này cho kết quả nén ảnh JPEG2000 đạt kết quả tốt nhất trên phần cứng bo mạch FPGA.

Trong trường hợp nén không tồn hao thì dữ liệu đi qua khối lượng tử (quantization) giữ nguyên so với đầu vào. Quá trình sắp xếp và đệm lại dữ liệu trong khối “quantization_buff” thì vẫn giống như nén có tồn hao.

Bảng 3. Giá trị mantissa và exponent

Mantissa $R_1=8$	Mantissa $R_1=16$	Exponent	level
14	22	1824	Level 5
14	22	1776	
14	22	1776	
14	22	1728	

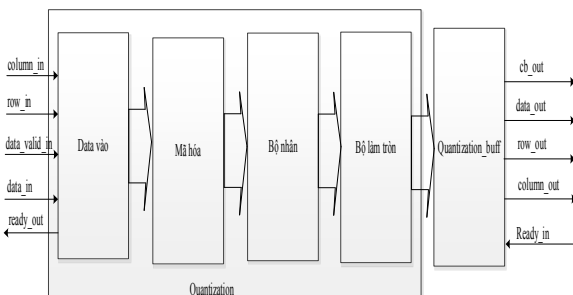
13	21	1824	Level 4
13	21	1792	
13	21	1792	
13	21	1760	
12	20	1848	Level 3
12	20	1872	
12	20	1872	
12	20	1896	
11	19	1874	Level 2
10	18	5	
10	18	5	
10	18	71	
9	17	36	Level 1
10	18	2003	
10	18	2003	
10	18	1890	

Phương pháp dùng trong lượng tử là uni-form scalar lượng tử có công thức như sau[1]:

$$q_b(i, j) = \text{sign}(y_b(i, j)) \left\lfloor \frac{|y_b(i, j)| \cdot W_i}{\Delta} \right\rfloor \quad (3)$$

Trong đó $(y_b(i, j))$ là hệ số của wavelet trong sub-band.

Δ là bước nhảy lượng tử trong phương trình (1) được xác định dựa vào các giá trị ϵ_b, R_b, μ_b để xác định bước nhảy lượng tử.



Hình 2. Sơ đồ khối thực hiện lượng tử hóa

Trong quá trình lượng tử phải thực hiện phép chia với bước nhảy trong từng sub-band. Trong thiết kế phần cứng việc thực hiện phép chia là rất phức tạp do phải sử dụng dấu chấm động nên tốn nhiều tài nguyên hệ thống. Để

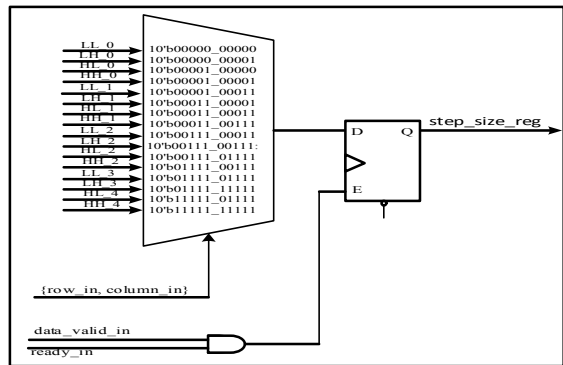
khắc phục vấn đề này trong bài báo thay vì thực hiện phép chia thì nhân nghịch đảo với với 8192, Kết quả phép nhân phải dịch lại 13 bit và làm tròn theo giải thuật “Banker’s rounding”. Ví dụ hàm “Banker’s rounding” làm tròn như sau:

2.5 làm tròn thành 2, với giá trị là 3.5 làm tròn thành 4. Ngoài ra các giá trị sau dấu chấm lớn hơn 0.5 được làm tròn thành 1 và nhỏ hơn 0.5 được làm tròn về 0.

Bảng 4 đã nhân nghịch đảo với với 8192.

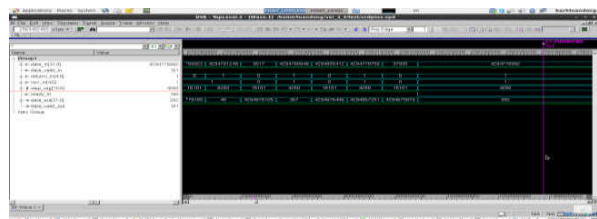
Bảng 4. Giá trị bước nhảy lượng tử

LEVEL	LL	LH, HL	HH
LEVEL=1	16100	8282	4260
LEVEL=2	34221	16344	7917
LEVEL=3	68900	34239	17015
LEVEL=4	138654	69905	35246
LEVEL=5	277309	280790	284359



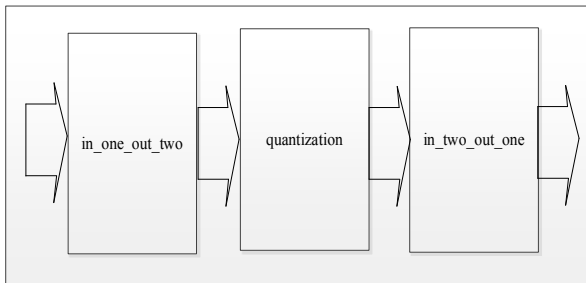
Hình 3. Bộ giải mã bước nhảy lượng tử hóa

Quá trình giải mã để chọn bước nhảy cho phù hợp với mỗi level. Để quá trình giải mã cần dựa vào hàng và cột trong khối FDWT. Giải mã bước nhảy lượng tử tương ứng trong Hình 3. Ta thấy trong Hình 4 row_in = 0, column_in = 0 thì bước nhảy lượng tử là 4260. Những giá trị bước nhảy lấy từ Bảng 4.



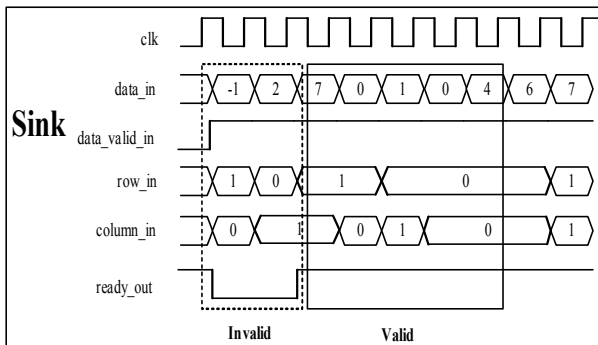
Hình 4. Giải mã tìm bước nhảy lượng tử

Ảnh GRAY (mức xám) biểu diễn 16-bit nén tổn hao. Quá trình lượng tử sẽ tràn bit trong level 4, 5 vì thực hiện phép chia. Để giải quyết vấn đề này chúng tôi đã tăng số bit cần biểu diễn dữ liệu lên tới 34-bit thay vì dùng 32-bit như ảnh GRAY 8-bit và ảnh GRAY 16-bit nhưng level nhỏ hơn 4, đồng thời thiết kế một khối tách dữ liệu và một khối gộp dữ liệu lại sơ đồ xem trong Hình 5. Vì dữ liệu ghi/đọc trong RAM chứa tối đa 32-bit để biểu diễn 1 dữ liệu, Nếu nhiều hơn thì tách ra một dữ liệu thành 2 dữ liệu có độ rộng bus nhỏ hơn hoặc bằng 32-bit.

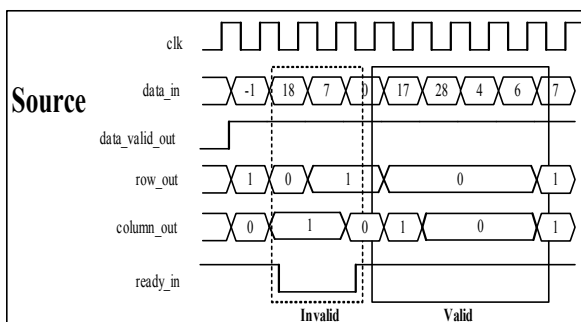


Hình 5. Thiết kế khối lượng tử khi GRAY 16 pixel level 4, 5

Trong thiết kế khối lượng tử cần phải tuân thủ theo chuẩn Avalon Streaming. Chuẩn này được quy định trong Hình 6, Hình 7.



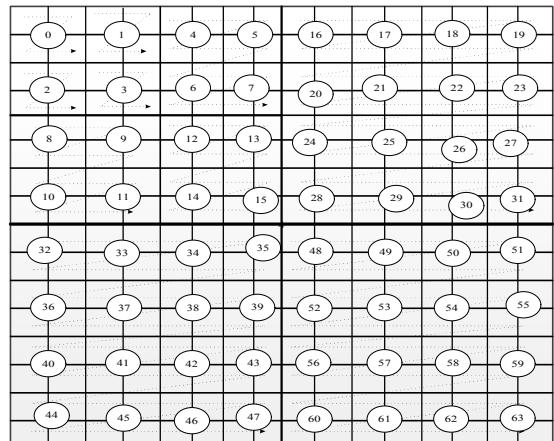
Hình 6. Giản đồ thời gian đầu vào khối lượng tử



Hình 7. Giản đồ thời gian đầu ra khối lượng tử

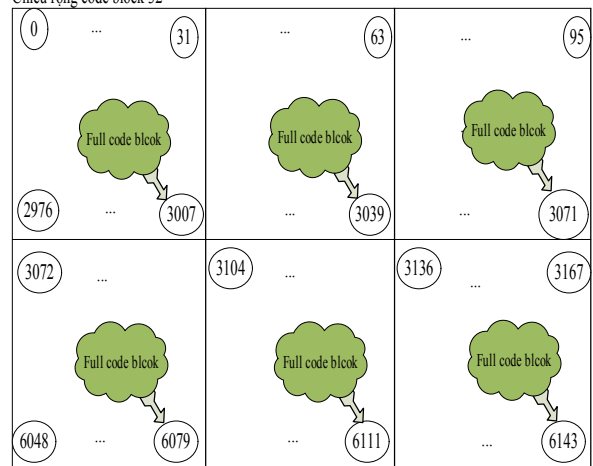
2.2 Thiết kế khối lượng tử đệm

Khối “quantization_buff” dùng để sắp xếp data theo từng vùng tần số riêng biệt tương ứng với từng sub-band khác nhau. Tiếp theo là đọc ra theo từng code-block. Dữ liệu được lưu trữ trong bộ nhớ. Khi đầy một code-block thì đọc dữ liệu ra. Cách thức quy định vị trí code-block thể hiện trong Hình 8



Hình 8. Quy ước đặt vị trí code-block trong một sub-band

Chiều dài là 96
 Chiều rộng là 64
 Chiều dài code block là 32
 Chiều rộng code block 32



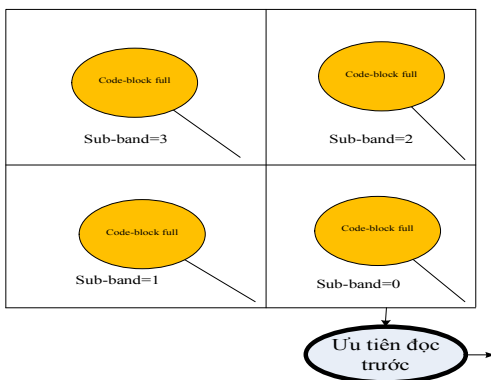
Hình 9. Vị trí bắt đầu và kết thúc một code-block trong sub-band

Trong Hình 9 là một ví dụ một sub-band có chiều dài 96 pixel, chiều rộng 64 pixel và chiều dài, rộng code-block là 32 pixel. Vị trí sub-band này lưu trong RAM được tính toán trước những vị trí đầy của mỗi code-block.

Khi ghi vào RAM thì tuần tự từ thấp đến cao. Khi đọc ra thì đọc từng code-block, đọc code-block nhỏ đến code-block lớn hơn, xem Hình 10. Cần phải tính toán được những vị trí mà code-block đầy để đọc ra. Nhưng nếu trong quá trình đó mà chưa đọc được thì chúng ta lại tiếp tục ghi vào, ghi đến khi nào đầy 2 band (có kích thước 2 lần chiều cao code-block) thì không được ghi nữa mà chờ đọc ra, để giải phóng bộ nhớ để tiếp tục ghi vào. Việc này được điều khiển bằng cách hạ ready_out xuống khi nào đọc xong lại bật ready_out lên tiến hành reset hết địa chỉ đọc và địa chỉ ghi tiến hành lại từ đầu.

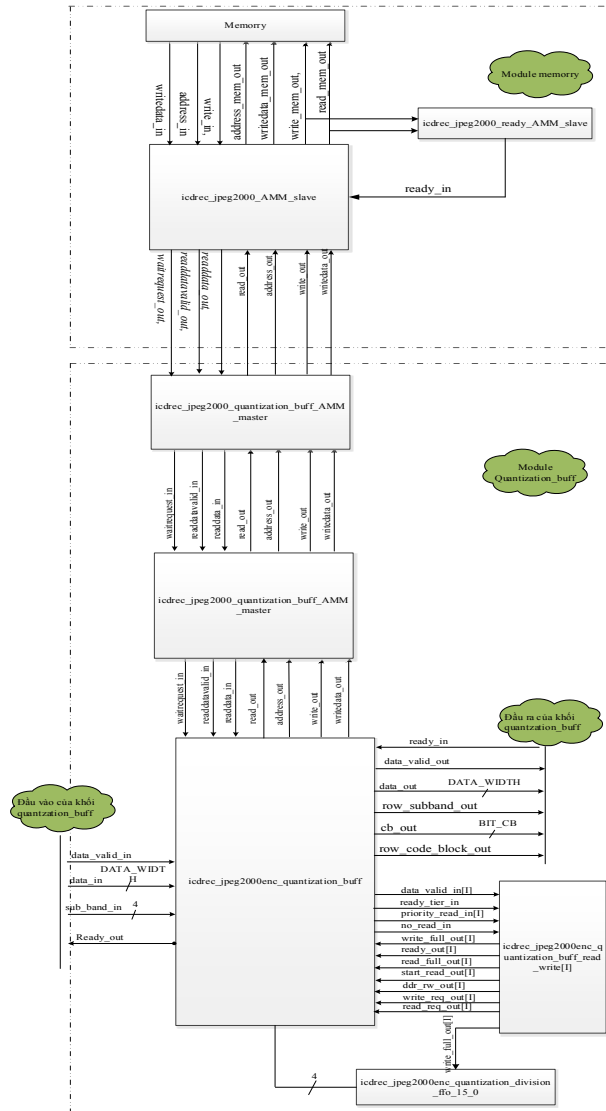
2.3 Mức độ ưu tiên của quá trình đọc ra

Trong trường hợp có nhiều vùng RAM đã chứa đầy dữ liệu thì quá trình ưu tiên đọc ra theo cách sau. Mức ưu tiên cao nhất là đọc code-block đang thực hiện. Duy trì trạng thái đọc dữ liệu trong code-block xong thì mới code-block khác. Thứ 2 là nếu nhiều code-block ở các sub-band khác nhau đang ở trạng thái đầy thì ưu tiên code-block ở vị trí sub-band nhỏ nhất trước minh họa Hình 10



Hình 10. Vùng dữ liệu ưu tiên đọc trước

Kiến trúc khối lượng tử đệm giao tiếp với DDR theo chuẩn Avalon Streaming kiến trúc được thể hiện trong Hình 11.



Hình 11. Sơ đồ kết nối khối lượng tử đệm với RAM ngoài

3. KẾT LUẬN

Trong bài báo này đã hỗ trợ lượng tử trong 2 chuẩn là nén tổn hao và không tổn hao. Trong quá trình không tổn hao thì kết quả hình ảnh nén có PSNR là inf (vô cùng).



Hình 12. Nén không tổn hao

Lượng tử không tổn hao được tổng hợp trong Bảng 5

Bảng 5. Kết quả nén không tổn hao

Kích thước hình	level	Dung lượng đầu	Dung lượng sau	PSNR
128x128	1	48.1KB	25.2 KB	inf
128x128	2	48.1KB	24.4 KB	Inf
256x256	3	192.1KB	141.2KB	Inf
512x512	4	768.1KB	352.6 KB	Inf
1024x1024	5	3.0 MB	812.7 KB	inf

Trong trường hợp nén không tổn hao thì kết quả PSNR khá cao hơn hơn 40 nên mắt người không phân biệt được. Thông thường thì mắt con người không phân biệt được hình nén và hình gốc nếu hình nén có chỉ số PSNR lớn hơn 37.



Hình 13. Nén có tổn hao

Bảng 6. Nén tổn hao

Kích thước hình	level	Dung lượng trước	Dung lượng sau	PSNR
128x128	1	48.1KB	20.2KB	51.54
128x128	2	48.1KB	19.2KB	51.73
256x256	3	192.1KB	118.7KB	51.63
512x512	4	768.1KB	245.6KB	51.91
1024x1024	5	3.0 MB	812.7KB	51.75

LỜI CẢM ƠN

Tác giả cảm ơn Trung Tâm Nghiên Cứu và Đào Tạo Thiết Kế Vi Mạch (ICDREC) đã cấp kinh phí cho tác giả nghiên cứu, thực hiện đề tài này và sự hỗ trợ của các đồng nghiệp. Bài báo là thành quả của đề tài mã số KC.01.13/11-15. Tác giả gửi lời cảm ơn chân thành đến Bộ Khoa học và Công nghệ, Văn phòng các chương trình trọng điểm cấp Nhà nước, Ban Chương trình KC.01 đã đầu tư kinh phí cho đề tài.

TÀI LIỆU THAM KHẢO

- [1] ISO/IEC 15444-1, *Information Technology-JPEG2000 Image Coding System*, Part 1: Core Coding System., 2000.
- [2] Tinku Acharya and Ping-Sing Tai, “*JPEG2000 Standard for Image Compression*”, Concepts, Algorithms and VLSI Architectures, New Jersey, U.S.A, 2005
- [3] Bach Tuan Dong, Ngo Thanh Dong, Nguyen Tuan Phuoc, Le My Ha “*decreasing capacity of JPEG2000 standard image on fpga board*” the 2nd international conference on green technology and sustainable development 2014. Ho Chi Minh City October.2014 pp.49-54.
- [4] Balster, E.J. Dept. of Electr. & Comput. Eng., Univ. of Dayton, Dayton, OH, USA Fortener, B.T. ; Turri, W.F. “*Integer computation of JPEG2000 wavelet transform and quantization for lossy compression*” IEEE Conference, July 2010 pp 495-500
- [5] Han oi’, Ali Bilgin’, and Michael W Marcellin l ‘*Department of Electrical and Computer Engineering, 2Biomedical Engineering The University of Arizona, Tucson, AZ, 85721, USA “Visibility thresholds for quantization distortion in JPEG2000” IEEE Conference, July 2009 PP 228 - 232*