

# NÉN ẢNH TRÊN FPGA

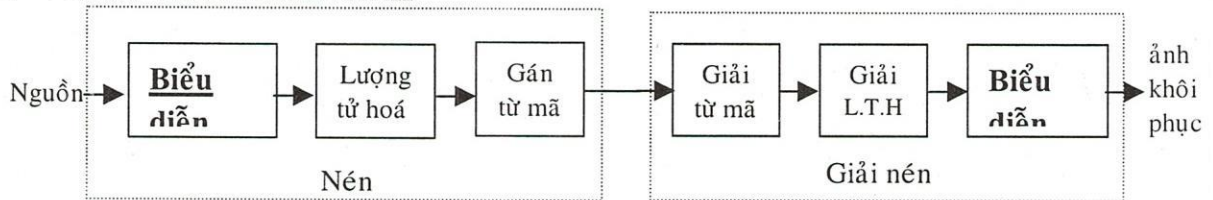
Hoàng Trang, Hồ Trung Mỹ

Khoa Điện-Điện Tử, Trường Đại Học Bách Khoa- ĐHQG TP.HCM

(Bài nhận ngày 24 tháng 2 năm 2005, hoàn chỉnh sửa chữa ngày 11 tháng 4 năm 2005)

**TÓM TẮT:** Bài báo này sẽ trình bày các hướng thực thi nén ảnh dựa vào DCT trên FPGA sao cho đạt được yêu cầu về thời gian nhanh, tài nguyên phần cứng tối ưu để đảm bảo được tính xử lý thời gian thực. Đồng thời, chúng tôi cũng cải tiến một phần giải thuật nén ảnh với mục tiêu tăng tốc độ tính toán, giảm bớt tài nguyên phần cứng. Cụ thể, chúng tôi thử nghiệm các mô hình thiết kế khác nhau trong từng khối của quá trình nén ảnh, tiến hành thử nghiệm trên họ SPARTAN2S của hãng Xilinx, tính toán thời gian thực hiện, tài nguyên phần cứng, công suất. Bên cạnh đó, chúng tôi quan sát và tính toán chất lượng ảnh thu được. Qua các thông số trên, so sánh thiết kế của chúng tôi với chuẩn JPEG.

## 1. NÉN VÀ GIẢI NÉN ẢNH



Hình 1. Mô hình hệ thống nén, giải nén

Ở khối đầu tiên của bộ nén, tín hiệu được trình bày dưới dạng thuận tiện để nén có hiệu quả nhất. Trong cách biểu diễn có hiệu quả, chỉ có phần nhỏ dữ liệu là cần thiết để truyền cho việc tái tạo lại tín hiệu.

Hoạt động thứ hai của bộ nén là lượng tử hóa, giúp rời rạc hóa thông tin được biểu diễn.

Hoạt động thứ ba là gán các từ mã. Các từ mã này là một chuỗi bit dùng để biểu diễn các mức lượng tử hóa.

Mỗi hoạt động cố gắng loại bỏ phần dư thừa trong tín hiệu và tận dụng sự giới hạn của hệ thống nhìn của mắt người. Nhờ bỏ đi các phần tử dư thừa, các thông tin giống nhau hoặc có liên quan đến nhau sẽ không được truyền đi. Những thông tin bỏ đi mà không ảnh hưởng đến việc nhìn, cũng sẽ không được truyền đi.

### 1.1. Biểu diễn thuận lợi

Trước tiên là quá trình chia ảnh thành nhiều khối ảnh nhỏ kích thước 4x4, 8x8 hoặc 16x16. Trong đó, chia thành các ảnh nhỏ kích thước 8x8 được chứng minh là tối ưu nhất [3]. Sau đó, tiến hành biến đổi DCT 2 chiều trên khối ảnh nhỏ này.

Tiến hành DCT 2 chiều trên X là một khối NxN điểm ảnh và tạo ra Y là một khối NxN hệ số DCT theo phương trình sau đây :

$$Y = A.X.A^T \quad \text{và biến đổi ngược IDCT : } X = A^T.Y.A$$

trong đó A là ma trận biến đổi với các hệ số

$$A_{ij} = C_i \cos \frac{(2j+1)i\pi}{2N} \quad \text{với } C_i = \sqrt{\frac{1}{N}} \quad (i=0), \quad C_i = \sqrt{\frac{2}{N}} \quad (i>0)$$

## 1.2. Lượng tử hóa

Bước tiếp theo của quá trình nén trong ảnh là lượng tử hoá các hệ số  $F(u,v)$  sao cho làm giảm được số lượng bit cần thiết.

Lượng tử hoá được thực hiện bằng việc chia các hệ số DCT 2 chiều  $F(u,v)$  cho các hệ số ở vị trí tương ứng trong bảng lượng tử  $Q(u,v)$ . Các hệ số có tần số thấp được chia cho các giá trị nhỏ, các hệ số ứng với tần số cao được chia cho các giá trị lớn hơn.

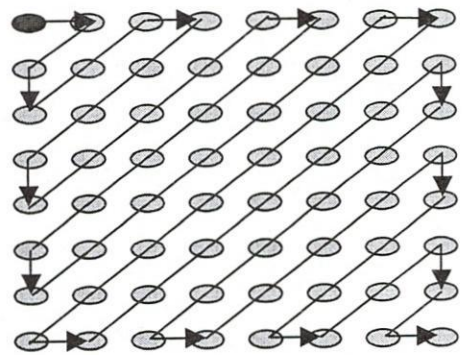
Kết quả ta nhận được bảng  $F_q(u,v)$  mới, trong đó phần lớn các hệ số có tần số cao sẽ bằng 0. Hệ số bảng lượng tử hoá thuận được xác định theo biểu thức:

$$F_q(u,v) = \left\lfloor \frac{F(u,v)}{Q(u,v)} \right\rfloor$$

Các giá trị  $F_q(u,v)$  sẽ được mã hoá trong các công đoạn tiếp theo.

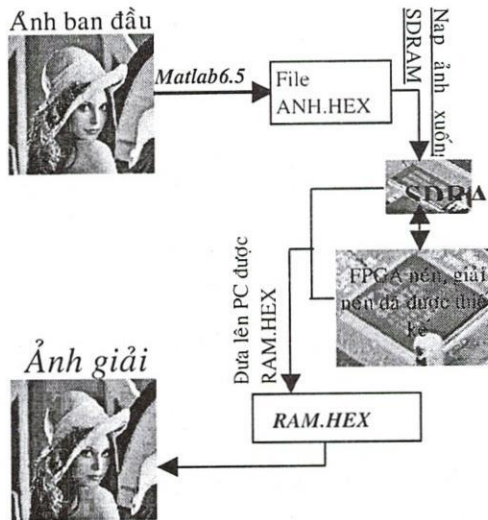
## 1.3. Mã hóa Entropy

Để mã hoá entropy các hệ số được lượng tử hoá  $C'(u,v)$ , trước hết cần biến đổi mảng hai chiều của các hệ số  $C'(u,v)$  thành chuỗi số một chiều bằng cách quét zig-zac.



Hình 2. Quét zig zắc các hệ số lượng tử hóa DCT

## 2. MÔ HÌNH THIẾT KẾ, THỬ NGHIỆM VÀ PHẦN CỨNG THỰC HIỆN



Hình 3. Mô hình thiết kế trên phần cứng

## 3. THIẾT KẾ

### 3.1. Thiết kế DCT

Chúng tôi thử nghiệm 4 hướng thiết kế DCT 2 chiều: theo hướng tuần tự, hướng song song, hướng hạn chế các trạng thái nghỉ và hướng đường ống (pipeline) tối ưu. Kết quả như sau:

Bảng 1: Các thông số			
DCT hướng tuần tự		DCT hướng hạn chế các trạng thái nghỉ	
Slice	1.151	Slice	1.614
Flip Flop	1.031	Flip Flop	1.547
LUT	1.988	LUT	2.983
Cổng tương đương	24.310	Cổng tương đương	36.020
Thời gian thực hiện (ns)	69066,23	Thời gian thực hiện (ns)	2327,25
Công suất tiêu thụ (mW)	500	Công suất tiêu thụ (mW)	750
DCT hướng song song		DCT hướng đường ống tối ưu	
Slice	1.726	Slice	1.251
Flip Flop	2.947	Flip Flop	1.040
LUT	3.153	LUT	2.094
Cổng tương đương	45.190	Cổng tương đương	25.940
Thời gian thực hiện (ns)	16806,48	Thời gian thực hiện (ns)	1148,7
Công suất tiêu thụ (mW)	870	Công suất tiêu thụ (mW)	510

Thiết kế lượng tử hóa

$$QDCT(i,j) = \frac{32 \times DCT(i,j)}{Matr\acute{a}n Q(i,j) \times tile\ luong\ tu} + k$$

trong đó:  $k = 1$  khi  $DCT(i,j) > 0$ ;  
 $= 0$  khi  $DCT(i,j) = 0$ ;  
 $= -1$  khi  $DCT(i,j) < 0$ .

Theo JPEG, RAM này lưu ma trận sau (hình 4)

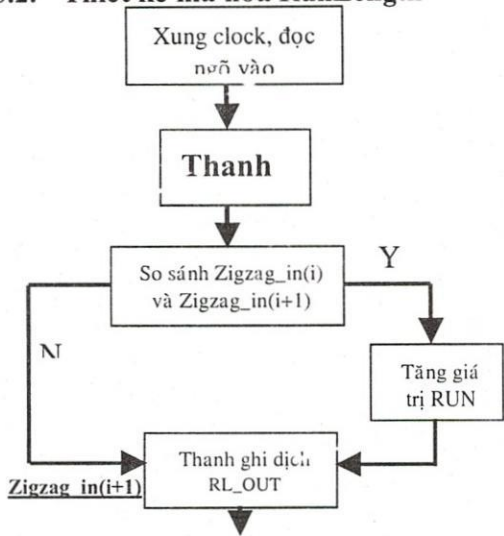
16 11 10 16 24 40 51 61	2 4 4 8 8 16 16 32
12 12 14 19 26 58 60 55	4 4 8 8 16 16 32 32
14 13 16 24 40 57 69 56	4 8 8 16 16 32 32 64
14 17 22 29 51 87 80 62	8 8 16 16 32 32 64 64
18 22 37 56 68 109 103 77	8 16 16 32 32 64 64 128
24 35 55 64 81 104 113 92	16 16 32 32 64 64 128 128
49 64 78 87 103 121 120 101	16 32 32 64 64 128 128 128
72 92 95 98 112 100 103 99	32 32 64 64 128 128 128 128

Hình 4: Ma trận lượng tử hóa  $Q(i,j)$  được thiết kế lưu trong RAM gồm 64 giá trị 8 bit

Theo phương thức cải tiến mà chúng tôi đề nghị, RAM này lưu ma trận sau:

	Lượng tử hóa theo JPEG	Lượng tử hóa hướng đề nghị
Slice	1.695	202
Flip Flop	1.262	171
LUT	3.199	968
Cổng tương đương	31.575	5.470
Thời gian thực hiện (ns)	213,56	102,54
Công suất (mW)	710	95

### 3.2. Thiết kế mã hóa RunLength



Slice	68
Flip Flop	81
LUT	100
Cổng tương đương	1.248
Thời gian thực hiện (ns)	21,62
Công suất (mW)	30

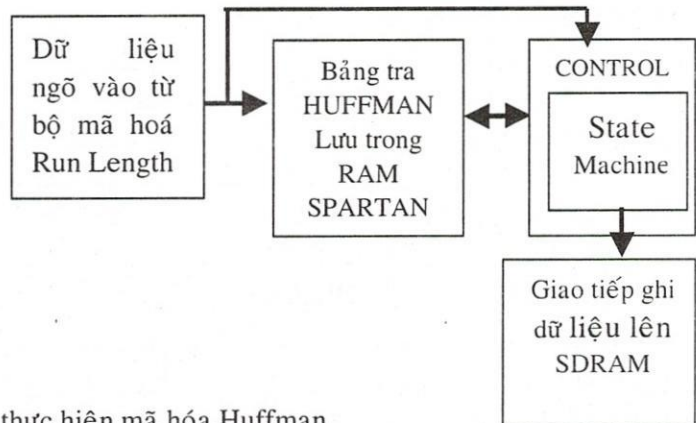
Hình 4. Sơ đồ thực hiện mã hóa RunLength

### 3.3. Thiết kế mã hóa Huffman

Mã hoá Huffman là giai đoạn cuối của quá trình nén ảnh, chúng tôi thiết kế lồng vào bộ mã hoá Huffman là máy trạng thái điều khiển cho việc ghi dữ liệu ảnh đã được nén lên SDRAM.

Mã hoá Huffman sẽ mã hoá một ký tự thành một chuỗi nhị phân chiều dài thay đổi [1]. Việc mã hoá này dựa trên xác suất xuất hiện của ký tự trong chuỗi dữ liệu. Để mã hoá, phải có một bảng mã, và nên thiết kế bảng mã này sẵn (lưu trong RAM của SPARTAN). Từ bảng mã này, chúng tôi sẽ tra tương ứng các ký tự.

Slice	120
Flip Flop	138
LUT	162
Cổng tương đương	2.062
Thời gian thực hiện (ns)	77,6
Công suất (mW)	45



Hình 5. Sơ đồ thực hiện mã hóa Huffman

## 4. PHÂN TÍCH KẾT QUẢ

Về vấn đề tài nguyên phần cứng, cần lưu ý về số transistor sử dụng, công suất tiêu thụ . . . Điều này sẽ ảnh hưởng đến vấn đề kích thước của chip ASIC được chế tạo, vấn đề kinh tế . . . Bên cạnh đó, về vấn đề thời gian, cần phải thiết kế sao cho phải tuyệt đối đảm bảo về vấn đề thời gian.

Chúng tôi xin phân tích một ví dụ sau :

Xét mỗi ảnh chứa 512x512 pixel; nghĩa là phải thực hiện  $\frac{512 * 512}{8 * 8} = 4096$  khối tính toán (với

một khối tính toán bao gồm thực hiện DCT, lượng tử hoá, quét ZigZag và mã hoá Huffman trên một khối 8x8 pixel; trong bài báo này, chúng tôi gọi là *khối đơn vị nén*) trên một ảnh. Mà trong một giây có 30 ảnh được truyền đi, nghĩa là cần thực hiện 4096\*30=122880 khối đơn vị nén/1s. Như vậy, một khối đơn vị nén cần thực hiện tối đa trong  $1.000.000.000ns/122880 = 8138ns$  .

Chúng tôi đưa ra công thức tính toán kích thước ảnh có thể nén được theo thời gian thực hiện của một khối đơn vị nén như sau (áp dụng đối với ảnh hình vuông):

$$N \cdot 10^4 \frac{1}{\sqrt{th.gian(ns) \cdot \delta}}$$


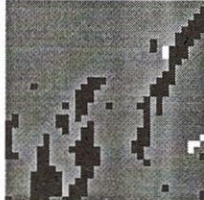


N là kích thước của ảnh con mà được chia nhỏ từ ảnh cần nén (N=4, 8, 16)  
 $\delta = 3; 2,5; 2,4$  khi chuẩn truyền video lần lượt là 30 frames/s, 25 frames/s, 24 frames/s.

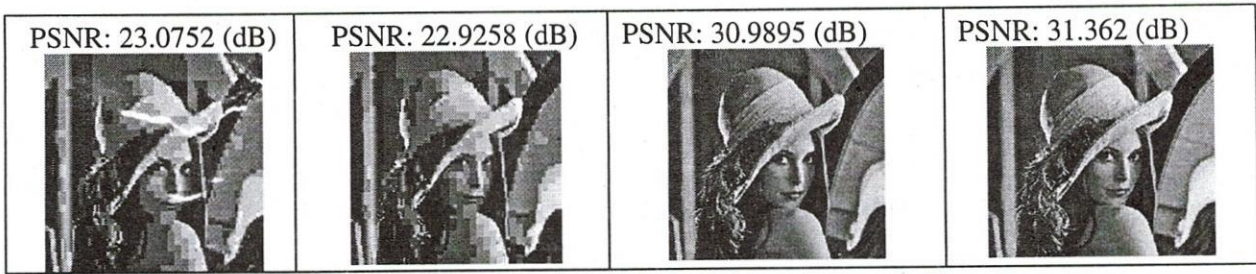
**Các thông số nén ảnh**

Ma trận theo chuẩn JPEG				
Cách thức	Tài nguyên (cổng)	Công suất (mW)	Thời gian tính cho 1 khối ảnh 8x8 (ns)	Kích thước ảnh (pixel xpixel)
Tuần tự	61.214	1.050	100615,38	145x145
Song song	83.582	1.300	20901,11	319x319
Hạn chế các trạng thái nghỉ	71.254	1.150	5539,33	620x620
Đường ống tối ưu	61.485	1.100	3671,02	762x762
Ma trận theo chuẩn hướng đề nghị				
Cách thức	Tài nguyên (cổng)	Công suất (mW)	Thời gian tính cho 1 khối ảnh 8x8 (ns)	Kích thước ảnh (pixel xpixel)
Tuần tự	33.445	750	79555,24	163x163
Song song	56.730	1.000	17683,92	347x347
Hạn chế các trạng thái nghỉ	45.550	950	4730,88	671x671
Đường ống tối ưu	34.050	750	2837,15	867x867

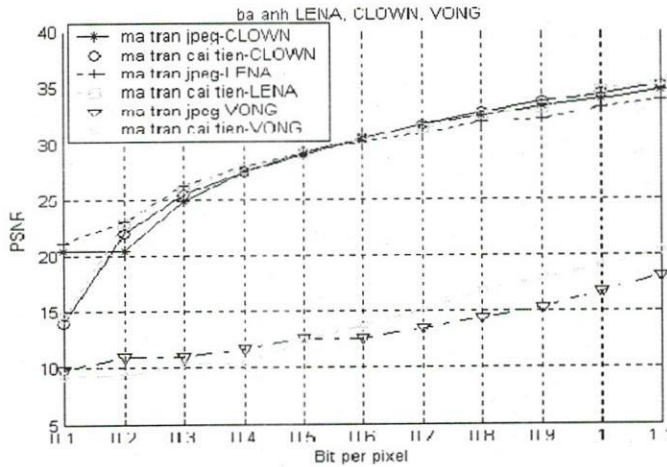
	Kích thước ảnh áp dụng được	
	Chuẩn JPEG	Ma trận đề nghị
<b>Bộ nén ảnh</b>	762x762	867x867
<b>Bộ giải nén</b>	702x702	817x817
<b>Hệ thống nén và giải nén</b>	702x702	817x817

Về chất lượng ảnh, chúng tôi thử nghiệm trên nhiều ảnh, từ các ảnh đơn giản cho đến các ảnh có nhiều chi tiết phức tạp, như sau đây là ảnh Lena thu được:

Chuẩn JPEG	Ma trận đề nghị	Chuẩn JPEG	Ma trận đề nghị
Cùng tốc độ bit 0,1 bpp		Cùng tốc độ bit 0,3 bpp	
PSNR: 21,1753 (dB)	PSNR: 14,4929 (dB)	PSNR: 26,3674 (dB)	PSNR: 25,9908 (dB)
			
Cùng tốc độ bit 0,2 bpp		Cùng tốc độ bit 0,7bpp	



Về chất lượng ảnh qua thông số PSNR, chúng tôi tính toán được qua phần mềm Matlab như sau:



Hình 6. Chất lượng ảnh qua thông số PSNR

### 5. KẾT LUẬN, HƯỚNG PHÁT TRIỂN

Vận dụng kỹ thuật đường ống vào trong hướng thiết kế ASIC là một hướng đi có thể mang lại nhiều hiệu quả. Kỹ thuật này giúp phát huy các tiềm năng, tăng tốc độ tính toán, đảm bảo các yêu cầu về tính thời gian thực, có thể áp dụng trong hệ thống thực.

Khi nén với tốc độ bit < 0,8 bpp, quan sát chúng tôi thấy có hiện tượng méo khối ở ảnh giải nén so với ảnh gốc. Điều này có thể giải thích như sau: hệ thống nén dựa trên DCT nén theo từng khối ảnh nhỏ, do đó không tận dụng được tương quan giữa các điểm ảnh tại các biên của các khối lân cận nhau. Chính lý do này cũng dẫn đến hệ thống chưa tận dụng hết hiệu suất nén.

Hệ thống nén, giải nén theo chuẩn JPEG và theo hướng đề nghị được so sánh một cách tổng quan như sau:

Kích thước ảnh		Chất lượng ảnh qua PSNR		Chất lượng ảnh qua quan sát	
		Bit rate < 0.5bpp	Bit rate ≥ 0.5bpp	Ảnh nhiều chi tiết phức tạp	Ảnh ít chi tiết phức tạp
JPEG	đề nghị	Hướng đề nghị xấu hơn chuẩn JPEG	Hướng đề nghị tốt hơn chuẩn JPEG	Hướng đề nghị xấu hơn chuẩn JPEG	Tương đương
762x762 (nén)	867x867 (nén)				
702x702 (giải nén)	817x817 (giải nén)				
702x702 (nén và giải nén)	817x817 (nén và giải nén)				

Khắc phục hiện tượng méo khối của thuật toán nén ảnh dựa trên DCT như biến đổi phủ lấp, Wavelets. Tuy nhiên, theo những thử nghiệm trên biến đổi Wavelets, chúng tôi nhận thấy thiết kế cấu trúc vi mạch cho nén ảnh dựa trên Wavelets rất tốn tài nguyên phần cứng. Vì vậy, một hướng nghiên

cứu mới là thiết kế cấu trúc vi mạch cho biến đổi Wavelets sao cho ít tốn tài nguyên phần cứng, đảm bảo yêu cầu về thời gian thực hiện . . .

## IMAGE COMPRESSION ON FPGA

Hoang Trang, Ho Trung My

Faculty of Electrical and Electronics Engineering, University of Technology-VNU HCM

**ABSTRACT:** *The article presents four methods for implementing the DCT-based image compression on FPGAs. The quantization matrix is applied to improve the performance of the approach. We have verified the implemetation of the approach by using the hardware SPARTAN2S of Xilinx Corporation. The hardware resource, timing as well as power consumption are also extracted from the experiment results. The proposed approach for image compression is also compared to the JPEG standard.*

## TÀI LIỆU THAM KHẢO

- [1]. David A. Huffman, "A Method for the Construction of Minimum Redundancy Codes", *Proceedings Of The I.R.E.*, 1972.
- [2]. David A.Patterson, John L.Hennessy, "Computer Architecture A Quantitative Approach", *Morgan Kaufmann*, ISBN 1-55860-372-7, 1996.
- [3]. Rafael C.Gonzalez, Richard E.Woods, "Digital Image Processing", *Addison-Wesley*, ISBN 0-201-50803-6, 1993.
- [4]. William K.Pratt, "Digital Image Processing", *John Wiley&Son*, ISBN 0-471-37407-5, 2001.
- [5]. Ling Guan, Sun-Yuan Kung, Jan Larsen, "Multimedia Image and Video Processing", *CRC*, ISBN 0-8493-3492-6, 2001.
- [6]. Kevin Skahill, "VHDL for Programable Logic", *Addison-Wesley*, ISBN 0-201-89586-2, 1996.
- [7]. Xilinx Corporation, manual "FPGA Editor Guide" , "Floorplanner Guide", "Timing Analyzer Guide".
- [8]. The ACM's First Electronic Publication, "Image compression-from DCT to Wavelets : A Review".
- [9]. Olof Larsson, Oscar Jonhansson, "Implementation of Motion-JPEG Using an ASIC Prototype Board", *Master's Thesis, Department of Information Technology*, 2001.
- [10]. Von Herrn Jorg Ritter, "Wavelet based image compression using FPGA", *Dissertation, University Halle-Wittenberg, Germany*, 2002.