

TỐI ƯU HÓA KIẾN TRÚC PHẦN CỨNG BỘ TẠO MA TRẬN ỨNG DỤNG CHO HỆ THỐNG LỌC BIÊN ẢNH TỐC ĐỘ CAO TRÊN FPGA

ARCHITECTURE OPTIMIZATION OF PIXEL-MATRIX GENERATOR FOR HIGH SPEED EDGE-DETECTOR BASED ON FPGA PLATFORM

Nguyễn Tuấn Phước, Nguyễn Minh Khánh Ngọc

Trung tâm Nghiên cứu và Đào tạo Thiết kế Vi mạch - ĐHQGTP.HCM

TÓM TẮT

Bài báo này trình bày phương pháp tối ưu phần cứng của bộ tạo ma trận nhằm thực hiện các khối xử lý ảnh tốc độ cao trên nền FPGA. Phương pháp này dựa vào cách bố trí các khối chức năng của FPGA để tối ưu hệ thống xử lý ảnh đáp ứng tốc độ xử lý dữ liệu thời gian thực. Bộ tạo ma trận sau khi được tối ưu đã được sử dụng trong hệ thống lọc biên ảnh tốc độ cao theo phương pháp Canny và trên chip FPGA giá rẻ.

Từ khóa: xử lý ảnh tốc độ cao, lọc biên ảnh, FPGA

ABSTRACT

This paper presents a method of optimization for pixel-matrix generator of high speed image-processing system on FPGA platform. This method is based on the layout of the function blocks in FPGA chip to optimize the system for real time processing. The optimized generator has been applied to design the high-speed edge detector using Canny algorithm and the low cost FPGA chip.

Key words: high speed image processing, edge detector, FPGA

I. GIỚI THIỆU

Xử lý biên trong xử lý ảnh số là một hệ thống con được xây dựng dựa trên các thành phần xử lý tín hiệu hai chiều, điển hình nhất là bộ lọc số hai chiều. Tuy nhiên, các điểm ảnh thường được truyền đi nối tiếp từng điểm một qua các bộ lọc số hai chiều. Điều này bắt buộc phải có một bộ tạo ma trận điểm ảnh hai chiều trước khi xử lý. Đối với xử lý bằng phần mềm, các điểm ảnh sẽ được lưu vào RAM và được đọc ra theo trật tự khác nhau nhằm tạo ra ma trận phù hợp. Đối với ảnh có độ phân giải lớn và tốc độ khung hình cao, chương trình phần mềm phải được thực hiện bằng CPU hoạt động ở tốc độ rất cao để có thể đáp ứng được yêu cầu của hệ thống. Nhằm giảm gánh nặng cho CPU, người thiết kế có thể sử dụng các GPU để thực hiện xử lý ảnh số dưới sự ràng buộc về tốc độ khung hình và độ phân giải ảnh. Phương pháp này làm giảm yêu cầu về tốc độ hoạt động của

CPU nhưng lại làm tăng chi phí của hệ thống do các GPU thường có giá thành cao. Một phương pháp khác là sử dụng chip FPGA thay thế GPU. Chip FPGA có tốc độ hoạt động không bằng GPU nhưng bù lại khả năng xử lý song song và khả năng tái cấu hình là ưu điểm vượt trội của loại chip này. Chip FPGA chứa các thành phần logic, các bộ nhớ RAM dạng cổng kép cho phép ghi đọc đồng thời trên cùng một cạnh xung đồng hồ và các thành phần DSP. Do FPGA cũng là một dạng ASIC, nên các thành phần của chip có vị trí cố định và tín hiệu được truyền giữa các thành phần thông qua mạng lưới dọc, ngang và cục bộ trong vùng nhỏ. Điều này sẽ làm giảm hiệu suất xử lý tín hiệu hai chiều của hệ thống nếu người thiết kế không cân bằng được vị trí các khối nhớ và khối xử lý dữ liệu. Để giải quyết vấn đề này chúng tôi đề xuất giải pháp chèn thêm thanh ghi sau các khối nhớ trong các FIFO để thực hiện tối

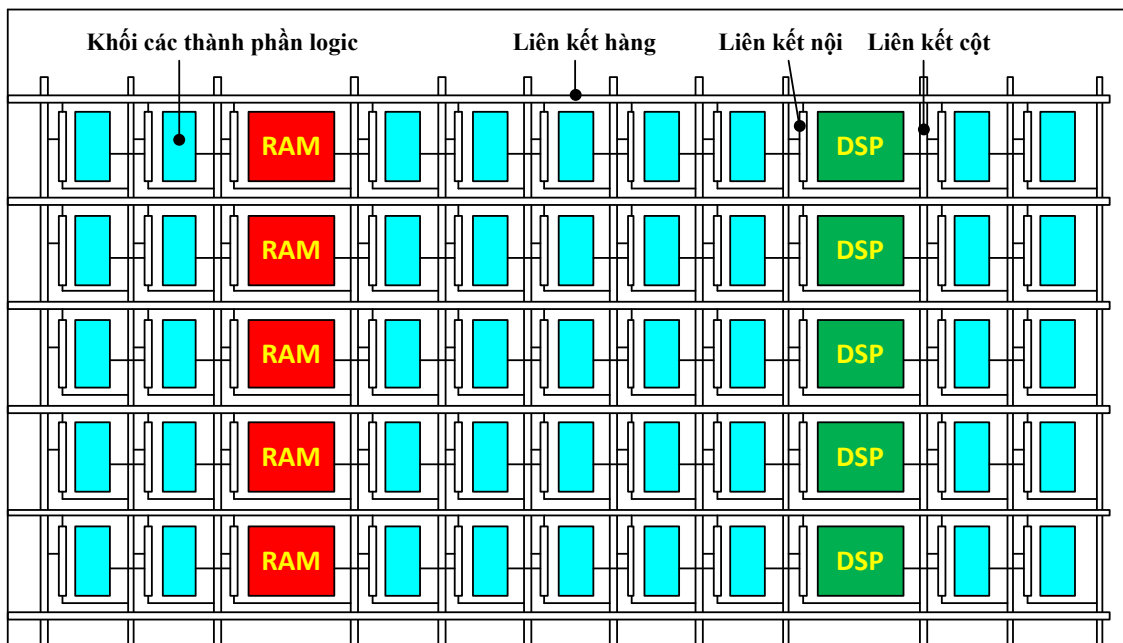
ưu tốc độ cho bộ tạo ma trận ứng dụng trong lọc biên ảnh tốc độ cao.

Nội dung của giải pháp được trình bày trong năm mục tiếp theo. Mục 2 trình bày về kiến trúc phần cứng của các chip FPGA, mục 3 trình bày về kiến trúc phần cứng của bộ tạo ma trận, mục 4 sẽ trình bày giải pháp chèn thanh ghi nhằm tối ưu tốc độ và mục 5 trình bày hệ thống xử lý biên ảnh theo phương pháp Canny đã ứng dụng bộ tạo ma trận sau khi thực hiện giải pháp tối ưu. Cuối cùng, phần kết luận được trình bày ở mục 6.

II. KIẾN TRÚC CHIP FPGA

Hiện nay, có rất nhiều loại chip FPGA được sử dụng để thử nghiệm các nguyên mẫu chip

trước khi sản xuất thử nghiệm. Tuy nhiên, các loại chip này có cấu trúc và các thành phần con tương tự nhau về mặt chức năng và bố trí phần cứng. Trong kiến trúc của FPGA [1, 2, 3, 4, 5], thành phần chính có chức năng tạo các mạch logic (cấu tạo của chúng bao gồm: một bảng tra tạo mạch tổ hợp và một D-FlipFlop để tạo mạch tuần tự) được đặt tên là LE (Logic Element) và được bố trí theo ma trận. Các LE được xếp thành các khối và các khối này được bố trí theo cột. Các cột LE được xếp song song với nhau. Các khối bộ nhớ hoặc các khối tính toán số học được chèn vào xen giữa những cột LE như trong Hình 1. Viên ngoài của chip FPGA là các khối quản lý tín hiệu liên kết bên ngoài chip được gọi là các “IO cell”.



Hình 1: Kiến trúc cơ bản của chip FPGA.

Để cung cấp các tín hiệu điện cho các thành phần trong chip FPGA, một mạng lưới các đường tín hiệu được thiết kế theo kiểu ma trận. Trong khuôn khổ một khối LE, các đường tín hiệu nội sẽ liên kết các LE trong khối. Các LE khác khối sẽ liên lạc với nhau qua các đường tín hiệu cột và hàng. Các khối LE có thể kết nối với các khối bộ nhớ hoặc các khối tính toán số học khi chúng được xếp kề cận nhau.

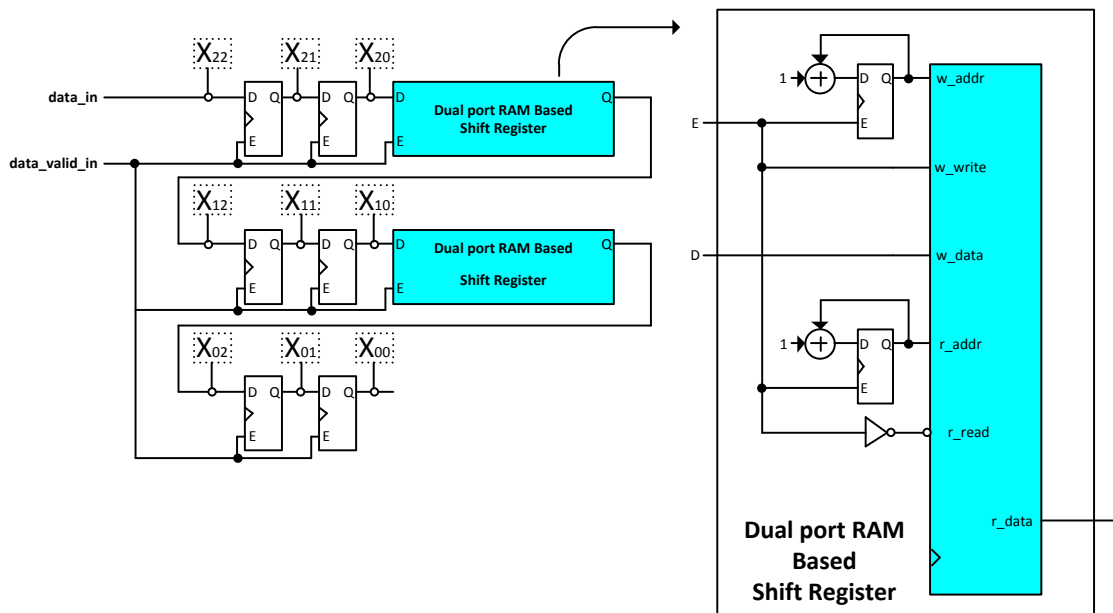
III. KIẾN TRÚC BỘ TẠO MA TRẬN

Trong xử lý ảnh, nhóm các điểm ảnh lân cận được xem là ngõ vào của bộ xử lý. Nhóm điểm ảnh lân cận này được chọn dưới hình thức của ma trận $M \times N$ (M hàng và N cột). Tập hợp các nhóm điểm ảnh này sẽ phủ đầy các ma trận ảnh cần xử lý; tương ứng với số lượng của nhóm điểm ảnh lân cận bằng với số lượng điểm ảnh của ảnh cần xử lý. Tuy nhiên, điểm ảnh được đưa vào khối

xử lý tuần tự từng điểm một. Điều này dẫn đến phải có một khối gom các điểm ảnh lân cận thành một nhóm như ma trận con $M \times N$. Kích thước của ma trận $M \times N$ này thường được chọn là số lẻ và bằng nhau (3x3, 5x5, 7x7,...). Khối này thường được gọi dưới tên là bộ tạo ma trận.

Trong thiết kế phần cứng, thanh ghi dịch sẽ đại diện cho chuỗi các điểm ảnh và tổ hợp vị trí truy xuất trong thanh ghi dịch sẽ tạo ra ma trận $M \times N$ mong muốn. Như vậy, để tạo nên một ma trận cơ bản 3x3, người thiết kế phải sử dụng một thanh ghi dịch có chiều

dài hơn gấp đôi chiều rộng của ảnh (tính với đơn vị điểm ảnh) ít nhất hai đơn vị. Điều này yêu cầu phải cần số lượng các D-FlipFlop rất lớn. Do đó, người thiết kế thường thay thế một phần thanh ghi dịch bằng bộ nhớ truy xuất ngẫu nhiên (RAM) có hai đầu ghi và đọc dữ liệu độc lập. Kiến trúc của một bộ tạo ma trận 3x3 được trình bày trong Hình 2. Thanh ghi dịch dựa trên bộ nhớ hai ngõ truy cập, hoạt động dựa vào bộ tạo địa chỉ ghi và địa chỉ đọc mỗi khi có tín hiệu cho phép dịch dữ liệu. Khoảng cách của địa chỉ ghi và địa chỉ đọc có giá trị đúng bằng số lượng thanh ghi con trong bộ thanh ghi dịch.



Hình 2: Kiến trúc bộ tạo ma trận 3x3

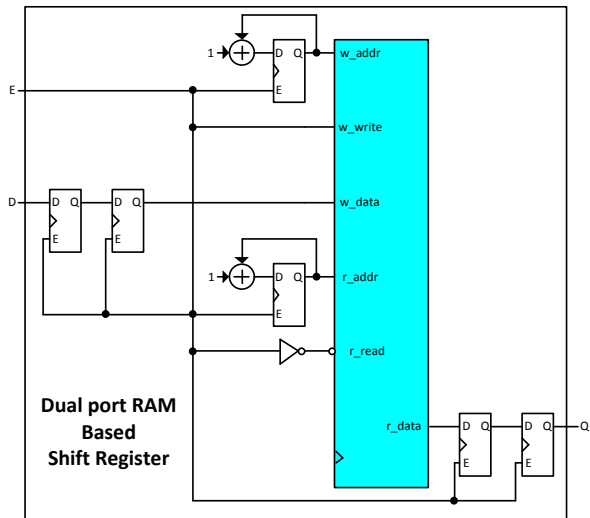
Như đã trình bày ở phần kiến trúc chip FPGA, giữa bộ nhớ và bộ phận tính toán số học chỉ có thể giao tiếp bằng các đường dữ liệu hàng và cột do vị trí của chúng trên chip cách xa nhau. Do đó khi dữ liệu ngõ ra từ khối tạo ma trận được đưa trực tiếp vào khối tính toán số học sẽ cần một đường truyền dữ liệu dài. Điều này làm giảm tần số xung đồng hồ của thiết kế và gây nên tình trạng thiết kế không đáp ứng được các hệ thống tốc độ cao.

IV. GIẢI PHÁP CÁCH LY KHỐI NHỚ VÀ KHỐI XỬ LÝ BẰNG THANH GHI

Như đã trình bày ở phần 3, vấn đề khoảng cách giữa bộ nhớ và bộ tính toán đã gây sự suy giảm tần số xung đồng hồ của thiết kế. Điều này làm cản trở thiết kế không thể đáp ứng yêu cầu tốc độ cao trong các hệ thống xử lý ảnh. Tuy nhiên, bố trí các thành phần logic nằm giữa các thành phần bộ nhớ và thành phần tính toán số học đã đưa ra một giải pháp để giải quyết vấn đề này. Các thành phần

logic có thể sử dụng liên kết nội để truyền dữ liệu cho các thành phần lân cận chúng. Như vậy, khi sử dụng thành phần logic làm các mắc xích liên kết thì trình tổng hợp phần cứng có thêm liên kết nội để tham gia giải quyết vấn đề.

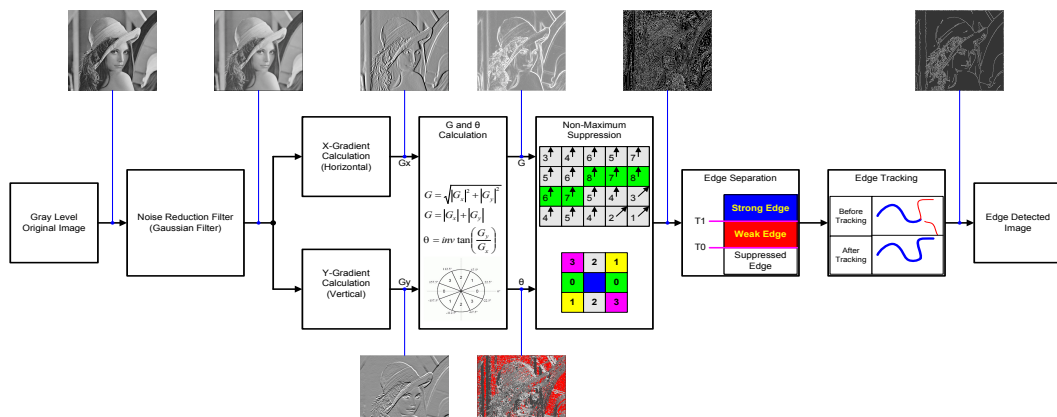
Tuy nhiên, số lượng thành phần logic được sử dụng trong một liên kết như vậy là một ẩn số mới. Nếu người thiết kế chọn ít quá thì tần số hoạt động của hệ thống sẽ chưa được đảm bảo. Nếu số lượng thành phần logic xen giữa này nhiều quá thì sẽ sinh ra hai vấn đề. Vấn đề thứ nhất là tốc độ thiết kế được đảm bảo nhưng lại gây ra hao phí về mặt tài nguyên không mong muốn. Vấn đề thứ hai là nếu sự hao phí tài nguyên lớn sẽ có dẫn đến trình tổng hợp không thể phân bố thiết kế số trên chip FPGA hiện tại hay phải thay thế chip hiện tại bằng chip có tài nguyên lớn hơn. Do đó, khi sử dụng giải pháp này, người thiết kế cần cân bằng giữa ba yếu tố: tốc độ chip FPGA, tốt độ thiết kế và hao phí tài nguyên của thiết kế. Thông số mặc định của chúng tôi đề nghị là hai thành phần logic ở ngõ ra của thanh ghi dịch dựa trên bộ nhớ hai ngõ truy cập như Hình 3. Trong các chip FPGA giá chỉ phí thấp, thì số lượng thành phần logic giữa khối bộ nhớ và khối tính toán số học thường nhỏ nhất là bốn. Do đó, hai là số lượng khá tốt để thực hiện giải pháp này.



Hình 3: Giải pháp chèn thêm thành phần logic cải thiện tốc độ

V. ỨNG DỤNG GIẢI PHÁP TỐI ƯU TRONG HỆ THỐNG LỌC BIÊN ẢNH TỐC ĐỘ CAO

Bộ tạo ma trận điểm ảnh cải thiện đã được áp dụng trên hệ thống lọc biên ảnh theo giải pháp Canny [6] (Hình 4) nhằm tạo nên một hệ thống lọc biên ảnh tốc độ cao với độ phân giải lớn. Hệ thống lọc biên ảnh theo phương thức Canny bao gồm sáu khối: Lọc nhiễu tuyến tính, lọc Sobel, tính biên độ và pha, khối loại các thành phần lân cận, phân loại cạnh và nối cạnh. Trong đó, ngoại trừ các khối tính biên độ và pha, các khối khác đều phải sử dụng bộ tạo ma trận để thực hiện các phép toán liên quan đến tham khảo điểm ảnh lân cận.



Hình 4: Kiến trúc hệ thống lọc biên ảnh theo phương thức Canny

Hệ thống được tổng hợp trên chip FPGA dòng Cyclone II (dòng chip chi phí thấp) của hãng Altera. Kết quả được liệt kê trong Bảng 1. Trong bảng kết quả tổng hợp, chúng tôi sử dụng ba dòng thiết bị có lượng tài nguyên khác nhau. Trong mỗi loại, lấy ra ba chip cùng tài nguyên nhưng tốc độ khác nhau

(hậu tố của mã hiệu chip càng lớn, tần số xung đồng hồ tối đa có thể đáp ứng được càng thấp). Kích thước ảnh được sử dụng trong hệ thống này bao gồm ba loại: HD 720 (1280x720), HD 1080 (1920x1080) và 2K (2048x1080).

Bảng 1: Bảng thống kê kết quả tổng hợp hệ thống lọc biên ảnh

STT	Kích thước ảnh	Tên thiết bị	Thành phần logic (LE) đã sử dụng	Bộ nhớ (bit) đã sử dụng	Bộ nhân nhúng 9 bit đã sử dụng	Tần số hoạt động tối đa (MHz)
1	1280x720	EP2C20F484C6	3664	115316	2	169.87
2	1280x720	EP2C35F672C6	3664	115316	2	180.67
3	1280x720	EP2C70F672C6	3664	115316	2	162.00
4	1280x720	EP2C20F484C7	3671	115316	2	138.41
5	1280x720	EP2C35F672C7	3672	115316	2	149.12
6	1280x720	EP2C70F672C7	3671	115316	2	141.80
7	1280x720	EP2C20F484C8	3669	115316	2	119.20
8	1280x720	EP2C35F672C8	3669	115316	2	120.34
9	1280x720	EP2C70F672C8	3669	115316	2	115.93
10	1920x1200	EP2C20F484C6	3711	172916	2	136.93
11	1920x1200	EP2C35F672C6	3713	172916	2	146.37
12	1920x1200	EP2C70F672C6	3714	172916	2	155.64
13	1920x1200	EP2C20F484C7	3715	172916	2	141.04
14	1920x1200	EP2C35F672C7	3716	172916	2	131.82
15	1920x1200	EP2C70F672C7	3715	172916	2	133.62
16	1920x1200	EP2C20F484C8	3711	172916	2	101.13
17	1920x1200	EP2C35F672C8	3715	172916	2	117.12
18	1920x1200	EP2C70F672C8	3716	172916	2	120.80
19	2048x1080	EP2C20F484C6	3774	184436	2	145.37
20	2048x1080	EP2C35F672C6	3716	184436	2	170.50
21	2048x1080	EP2C70F672C6	3715	184436	2	150.74
22	2048x1080	EP2C20F484C7	3775	184436	2	144.59
23	2048x1080	EP2C35F672C7	3717	184436	2	144.05
24	2048x1080	EP2C70F672C7	3715	184436	2	115.83
25	2048x1080	EP2C20F484C8	3772	184436	2	107.82
26	2048x1080	EP2C35F672C8	3718	184436	2	116.04
27	2048x1080	EP2C70F672C8	3718	184436	2	103.84

Tốc độ đáp ứng thấp nhất xấp xỉ 100MHz tương ứng với 45 fps cho kích thước 2048x1080 hay 108 fps cho kích thước 1280x720. Như vậy, hệ thống lọc biên ảnh đã đạt yêu cầu đáp ứng lọc biên cho ảnh có độ phân giải lớn và tốc độ khung cao.

VI. KẾT LUẬN

Giải pháp chèn thêm các thành phần logic vào giữa liên kết của khối bộ nhớ và khối

tính toán số học đã cải thiện được tốc độ của các thiết kế xử lý ảnh tốc độ cao trên nền chip FPGA chi phí thấp. Tuy nhiên, người thiết kế cần lưu ý cân bằng giữa việc cải thiện tốc độ đáp ứng, tốc độ chip FPGA và hao phí tài nguyên khi bắt tay vào chọn lựa số lượng thành phần logic được chèn vào liên kết.

TÀI LIỆU THAM KHẢO

- [1] Altera, Cyclone II Device Handbook, 2008.
- [2] Altera, Cyclone III Device Handbook, 2012.
- [3] Altera, Cyclone IV Device Handbook, 2014.
- [4] Xilinx, Spartan-3 Generation FPGA User Guide, 2011.
- [5] Xilinx, Spartan-6 FPGA Configurable Logic Block, 2012.
- [6] J. CANNY, "A computational approach to edge detection," IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, vol. VOL. PAMI-8, pp. 679–698, 1986.