ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

Nguyễn Khâm Hồng Quang

NGHIÊN CỨU PHÁT TRIỂN HỆ THỐNG PHẦN CỨNG, PHẦN MỀM PHÁT HIỆN VÀ THEO DÕI CHUYỂN ĐỘNG TRÊN CƠ SỞ CÔNG NGHỆ FPGA

LUẬN VĂN THẠC SĨ ĐIỆN TỬ VIỄN THÔNG
Ngành: Kỹ thuật Điện tử

HÀ NỘI – 2020

ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

Nguyễn Khâm Hồng Quang

NGHIÊN CỨU PHÁT TRIỂN HỆ THỐNG PHẦN CỨNG, PHẦN MỀM PHÁT HIỆN VÀ THEO DÕI CHUYỂN ĐỘNG TRÊN CƠ SỞ CÔNG NGHỆ FPGA

LUẬN VĂN THẠC SỸ ĐIỆN TỬ VIỄN THÔNG
Ngành: Kỹ thuật Điện tử

Giảng viên hướng dẫn: PGS.TS Trần Xuân Tú

HÀ NỘI – 2020

LỜI CAM ĐOAN

Tôi xin cam đoan rằng luận văn tốt nghiệp này hoàn toàn là công trình nghiên cứu thực sự của tôi, được thực hiện dựa trên cơ sở nghiên cứu lý thuyết, kiến thức ngành và chuyên ngành, nghiên cứu khảo sát tình hình thực tiễn và dưới sự hướng dẫn của PGS.TS. Trần Xuân Tú.

Các số liệu, bảng biểu và những kết quả trong luận văn tốt nghiệp là trung thực, những nhận xét, kết luận được đưa ra xuất phát từ thực tiễn thực nghiệm và kinh nghiệm của bản thân tôi.

Những phần sử dụng tài liệu tham khảo trong khóa luận tốt nghiệp đã được nêu rõ ở mục "Tài liệu tham khảo", nếu có điều gì không trung thực tôi xin nhận hoàn toàn trách nhiệm về mình và chịu mọi hình thức kỷ luật của khoa và nhà trường.

Tác giả luận văn thạc sĩ:

Nguyễn Khâm Hồng Quang

TÓM TẮT

***Tóm tắt*:** Ngày này công nghệ xử lý ảnh đang ngày càng phát triển và đưa vào ứng dụng trong rất nhiều các thiết bị di động, ứng dụng trong các hệ thống camera để đảm bảo an ninh công cộng. Ngoài ra, xử lý ảnh còn xuất hiện trong rất nhiều ứng dụng khác trong quốc phòng, trật tự trị an và giám sát các công việc độc hại. Có thể thấy rằng công nghệ càng ngày càng được sử dụng để để phục vụ con người ở những mục đích khác nhau. Tuy nhiên, các công nghệ xử lý ảnh ngày nay chủ yếu được thực hiện trên phần mềm. Do đó, không thể cải thiện được hiệu năng xử lý ảnh cũng như chỉnh sửa để phù hợp với nhu cầu sử dụng của từng người. Dựa trên sự việc phân tích các yêu cầu thực tế đặt ra. Đề tài này nghiên cứu về một hệ thống phần cứng, phần mềm về xử lý ảnh dựa trên công nghệ FPGA để áp dụng thực thi các thuật toán xử lý ảnh về sau từ đó đưa vào các ứng dụng ra đời sống.

Từ khóa: *FPGA, hệ thống xử lý ảnh số.*

DANH SÁCH CÁC THUẬT NGỮ VÀ TỪ VIẾT TẮT

Viết tắt	Tiếng Anh	Tiếng Việt
ALU	Arithmetic Logic Unit	Bộ thực thi các phép toán lô-gic
ALU	Arithmetic Logic unit	Đơn vị lô-gic số học
AMBA	Advanced Microcontroller Bus Architecture	Kiến trúc bus của vi điều khiển tiên tiến
ASIC	Application Specific Integrated Circuit	Vi mạch chuyên dụng
BRAM	Block Random Access Memory	Khối bộ nhớ truy cập ngẫu nhiên
CPLD	Complex Programmable Logic Device	Cấu kiện lô-gic khả trình phức tạp
CPU	Central Processing Unit	Đơn vị xử lý trung tâm
EEPROM	Electrically Erasable ROM	ROM lập trình được và xóa được bằng điện
FPGA	Field Programmable Gate Array	Mảng cổng lô-gic có khả năng lập trình
DL	Hardware Description Language	Ngôn ngữ mô tả phần cứng
IC	Intergrated Circuits	Mạch tích hợp
LUT	Look Up Table	Bảng tra cứu
PAL	Programmable Array Logic	Mảng lô-gic lập trình được
PLA	Programmable Logic Array	Vi mạch lập trình dùng mảng lô-gic
PLD	Programmable logic devices	Thiết bị lô-gic lập trình được
PROM	Programmable ROM	ROM khả trình
RAM	Random Access Memory	Bộ nhớ truy cập ngẫu nhiên
RISC	Reduced Instruction Set Computer	Máy tính có kiến trúc tập lệnh thu gọn
ROM	Read Only Memory	Bộ nhớ chỉ đọc
RTL	Register Transfer Language	Ngôn ngữ chuyển dịch thanh ghi
SPLD	Simple Programmable Logic Device	Cấu kiện lô-gic khả trình đơn giản
SRAM	Static RAM	Bộ nhớ truy cập ngẫu nhiên tĩnh
VHDL	Very High Speed Hardware Description Language	Ngôn ngữ mô tả phần cứng cho các mạch tích hợp tốc độ rất cao

DANH SÁCH HÌNH ẢNH

DANH SÁCH BẢNG

GIỚI THIỆU

Ngày nay, xử lý ảnh đóng vai trò hết sức quan trọng trong đời sống hàng ngày. Các ứng dụng về xử lý ảnh được đưa vào sử dụng trong các thành phố thông minh, hệ thống giám sát an ninh, hệ thống giao thông và trong an ninh quốc phòng… ngày càng trở nên phổ biến. Tuy nhiên, hầu hết các hệ thống camera hiện nay đều thực thi các thuật toán xử lý ảnh thông qua các phần mềm chạy trên nền một hệ thống vi xử lý phức tạp. Việc này hạn chế năng lực xử lý ảnh cũng như gây tiêu hao công suất tiêu thụ khi một số lượng lớn camera tích hợp được trong cùng một hệ thống giám sát. Một trong những yêu cầu về mặt công nghệ là cứng hóa một số thuật toán xử lý ảnh để nâng cao hiệu suất xử lý cũng như giảm tải áp lực cho các vi xử lý trên thiết bị (có nghĩa là kết hợp giữa phần cứng và phần mềm trong việc thực thi các thuật toán xử lý ảnh của các thiết bị giám sát hình ảnh). FPGA là một mạch tích hợp cỡ lớn dùng cấu trúc mảng phần tử lô-gic mà người dùng có thể lập trình và thay đổi thiết kế của mình. Kiến trúc phần cứng của mảng cổng lô-gic có thể lập trình cho phép tích hợp số lượng tương đối lớn các phần tử bán dẫn vào một vi mạch. Ngoài ra mảng cổng lô-gic có khả năng lập trình là một công nghệ đang được ứng dụng rộng rãi vì khả năng tái cấu hình và chi phí cũng rẻ hơn so với một số loại mạch tích hợp khác. Nó phù hợp cho việc tạo ra một hệ thống mẫu để sau này đưa vào sản xuất. Trong đề tài này sẽ có ba phần chính là tìm hiểu về một hệ thống và các bước xử lý ảnh số cơ bản hiện nay, tìm hiểu về công nghệ FPGA và kiến trúc của nó, tích hợp thuật toán xử lý ảnh lên hệ thống FPGA cụ thể là bo mạch Artix-7 AC701 và mô phỏng đưa ra một số kết quả đã thực thi được. Luận văn được thực hiện tại Phòng thí nghiệm trọng điểm Hệ thống tích hợp thông minh trong khuôn khổ đề tài QG.18.38 và đề tài 28/2018/ĐTĐL.CN-CNC.

Nội dung luận văn sẽ gồm các chương sau:

Chương 1: Hệ thống xử lý ảnh số: Giới thiệu về xử lý ảnh số và các thành phần hệ thống xử lý ảnh số, các thuật toán xử lý ảnh về theo dõi chuyển động, xác định đối tượng và một số giải pháp phần cứng thực thi một hệ thống xử lý ảnh.

Chương 2: Công nghệ FPGA: Tìm hiểu về các loại mạch tích hợp, kiến trúc của mạch tích hợp FPGA, giới thiệu về Artix-7.

Chương 3: Đề xuất và thực thi thử nghiệm hệ thống xử lý ảnh trên FPGA

Cuối cùng là phần kết luận: Tổng kết những gì đã làm được và đề xuất phương hướng phát triển.

Chương 1

HỆ THỐNG XỬ LÝ ẢNH SỐ

Ngày nay, xử lý ảnh đang là một lĩnh vực khoa học công nghệ được đầu tư và phát triển một cách nhanh chóng và rộng rãi. Nó được ứng dụng rất rộng rãi trong các thiết bị điện tử phục vụ cho đời sống hàng ngày cũng như trong các lĩnh vực quan trọng trong quân sự, nhà máy, xí nghiệp lớn … Đi cùng với sự phát triển của các thuật toán xử lý ảnh nhanh thì cùng với đó là một hệ thống phải thực thi được một khối lượng tính toán vô cùng lớn và tốc độ xử lý phải nhanh để có thể khai thác tối đa các ứng dụng mà xử lý ảnh mang lại. Sau đây học viên sẽ giới thiệu về xử lý ảnh và các khái niệm cơ bản của một hệ thống xử lý ảnh số.

1.1 Giới thiệu về xử lý ảnh số

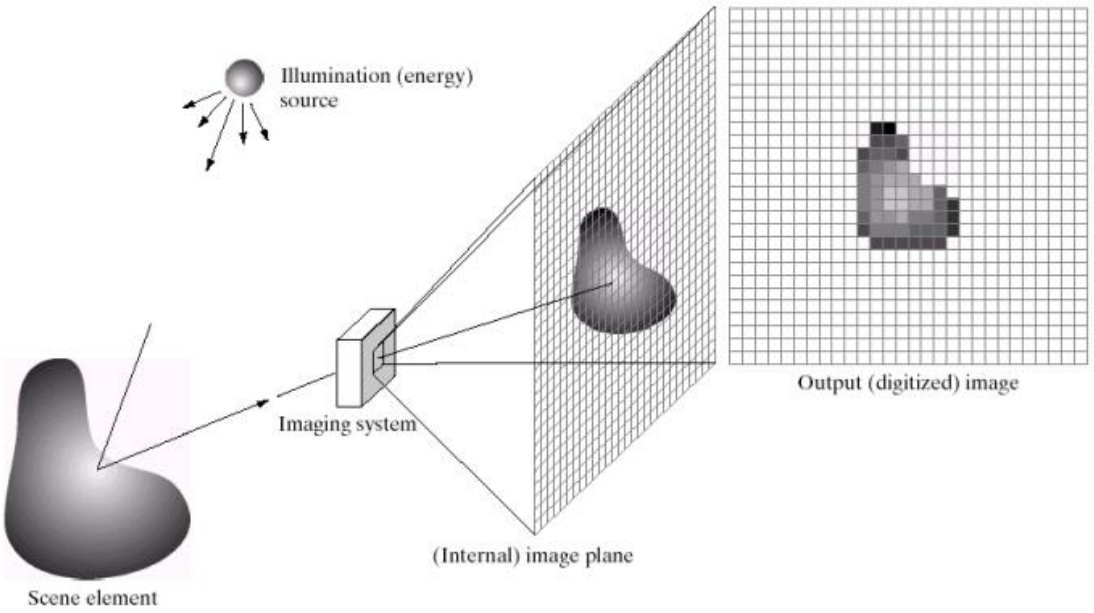
Xử lý ảnh là quá trình biến đổi ảnh ban đầu thành một ảnh mới với các đặc tính tuân theo mục đích của người sử dụng. Xử lý ảnh bao gồm các quá trình phân tích, phân lớp các đối tượng làm tăng chất lượng, phân đoạn và tách cạnh, gán nhãn cho vùng hay quá trình biên dịch các hình ảnh thông tin của ảnh.

Cũng như xử lý dữ liệu bằng đồ họa, xử lý ảnh số là một lĩnh vực của tin học ứng dụng. Xử lý dữ liệu bằng đồ họa đề cập đến những ảnh nhân tạo, các ảnh này được xem xét như là một cấu trúc dữ liệu và được tạo bởi các chương trình. Công nghệ xử lý ảnh số gồm các phương pháp và kỹ thuật biến đổi ảnh, để từ đó truyền tải hoặc mã hóa các ảnh tự nhiên. Kết quả đầu ra của xử lý ảnh chính là một ảnh mới tốt hơn hoặc một kết luận từ các thuộc tính đầu vào. Mục đích của xử lý ảnh số bao gồm [12]:

- Biến đổi ảnh làm tăng chất lượng ảnh.
- Tự động nhận dạng đối tượng, phát hiện, đánh giá các nội dung của ảnh.

Ảnh trong thực tế là tập hợp rất nhiều các điểm ảnh mà trong đó mỗi điểm ảnh có các giá trị đặc trưng về độ sáng hay một đặc tính nào đó tại một vị trí ngẫu nhiên của đối tượng

trong một vùng không gian và nó có thể xem như một hàm n biến P($C_1, C_2, C_3, \dots, C_n$). Do đó ảnh trong thực tế có thể xem như ảnh n chiều. Tuy nhiên khi đưa vào máy tính thì ảnh được xử lý thành một hình ảnh hai chiều với tập hợp các điểm ảnh có giá trị khác nhau đó được gọi là ảnh số như Hình 1.1:



Hình 1.1: Ảnh số [5].

Có ba mức độ trong xử lý ảnh số được liệt kê trong bảng 1.1:

Bảng 1.1: Các cấp độ trong xử lý ảnh

Xử lý mức thấp	Xử lý mức vừa	Xử lý mức cao
Đầu vào: Ảnh Đầu ra: Ảnh Kết quả: Loại bỏ các nhiễu đầu vào, làm sắc nét hình ảnh hơn	Đầu vào: Ảnh Đầu ra: Các thuộc tính Kết quả: Nhận dạng đối tượng, phân vùng ảnh	Đầu vào: Các thuộc tính Đầu ra: Hiểu được nội dung của các thuộc tính và đưa ra kết quả phù hợp. Kết quả: Hiểu được bối cảnh, tự điều hướng.

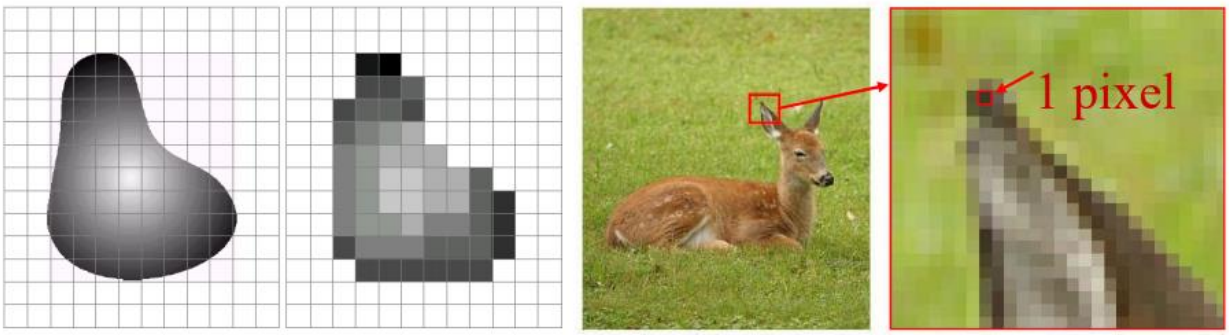
1.2 Các quá trình xử lý ảnh số

1.2.1 Một số khái niệm cơ bản trong xử lý ảnh số

Muốn đi vào phân tích các bước xử lý ảnh số chúng ta cần biết một số khái niệm cơ bản trong xử lý ảnh số.

Ảnh số: Ảnh số là tập hợp các điểm ảnh với các mức độ sáng phù hợp dùng để mô tả ảnh gần với ảnh thật nhất.

Điểm ảnh: Trong quá trình số hóa ảnh để biến đổi một ảnh liên tục thành một tập điểm phù hợp với ảnh thật về vị trí và độ sáng. Khoảng cách giữa các điểm đó được thiệt lập sao cho mắt thường không nhìn thấy được ranh giới giữa các điểm. Mỗi điểm như vậy trong một ảnh số được gọi là điểm ảnh hay gọi tắt là pixel. Trong một bức ảnh hai chiều mỗi pixel tương ứng với một điểm trong hệ tọa độ (x, y) được thể hiện trong Hình 1.2:



Hình 1.2: Điểm ảnh (pixel) [5].

Độ phân giải của ảnh: Độ phân giải của ảnh là mật độ các điểm ảnh được thiết lập trên ảnh hiển thị. Việc lựa chọn khoảng cách thích hợp giữa các điểm ảnh để mắt thường không nhìn thấy được chính là độ phân giải và nó cũng được phân bố theo trục (x, y) trong không gian hai chiều. Một ví dụ cụ thể như độ phân giải 1920*1280 chính là trong một khung hình có 1920 điểm ảnh chiều ngang và 1280 điểm ảnh chiều dọc. Lý do vì vậy trong một khung hình độ phân giải càng cao thì độ mịn của ảnh càng lớn vì khi độ phân giải càng cao thì khoảng cách giữa các điểm ảnh trong một khung hình càng bé.

Mức xám của ảnh: Mức xám của ảnh chính là cường độ sáng của điểm ảnh đó được gán bằng các giá trị số tại điểm đó. Có rất nhiều thang mức xám cơ bản được giới hạn bởi số bit trong một điểm ảnh. Số bit càng lớn tương đương với việc ảnh có độ chính xác so với ảnh gốc càng tăng. Cùng với đó là ba loại ảnh cơ bản đại diện cho các mức xám khác nhau.

Ảnh xám (gray image): là ảnh chỉ có hai màu đen trắng tuy nhiên mức xám ở các điểm ảnh có thể khác nhau. Ảnh này mỗi điểm ảnh có giá trị 8bit tương ứng với có 256 mức sáng của điểm ảnh.

Ảnh nhị phân: là ảnh chỉ có hai mức sáng cơ bản là đen và trắng và chỉ được biểu diễn bởi 1 bit với bit 0 là màu trắng và bit 1 là màu đen. Ảnh nhị phân có thể được biểu diễn bằng ảnh xám với độ sáng tương ứng là 0 và 255.

Ảnh màu: Với ảnh màu thì mỗi điểm ảnh được biểu diễn bởi 3 màu cơ bản là đỏ, xanh lục, xanh dương. Mỗi màu có giá trị độ sáng từ 0-255 cho nên để biểu diễn ảnh màu mỗi điểm ảnh cần 3 bytes để biểu diễn.

Ngày nay với sự phát triển trong công nghệ xử lý ảnh đã phát triển lên những ảnh có giá trị lớn hơn 3 bytes để biểu diễn một điểm ảnh. Ngoài các giá trị màu cơ bản người ta còn đưa vào các yếu tố khác để xử lý chất lượng ảnh được tốt hơn.

Ảnh xám Ảnh nhị phân. Ảnh màu.

Hình 1.3: Các loại ảnh trong xử lý ảnh.

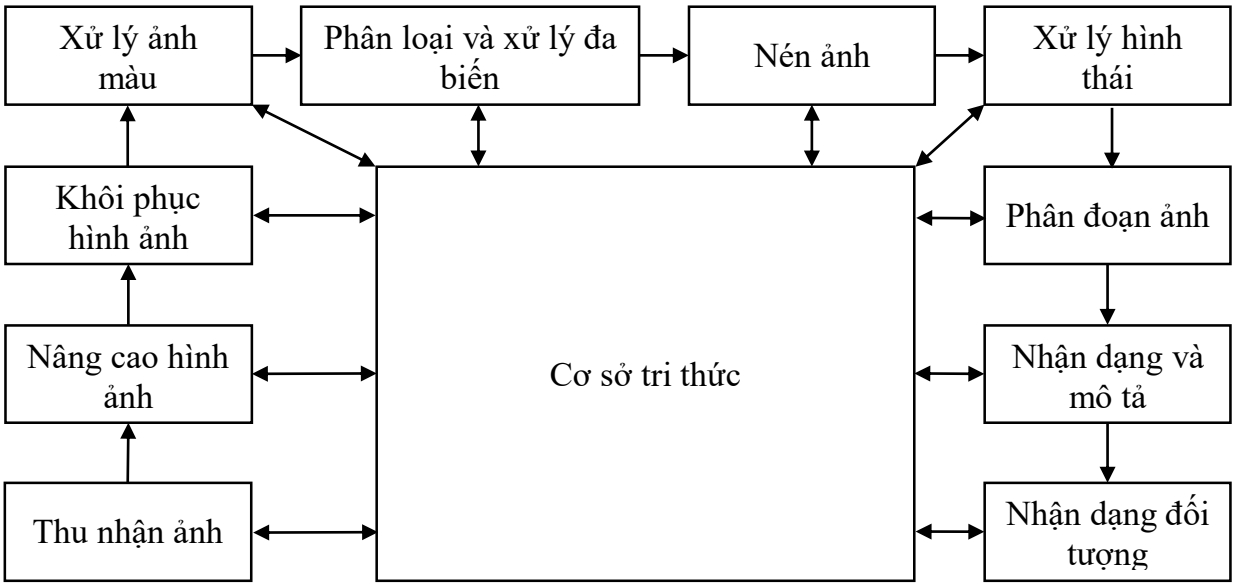
1.2.2 Các bước xử lý ảnh số

Có 10 bước cơ bản trong xử lý ảnh số được thể hiện trong hình 1.4 dưới đây:

Thu nhận ảnh: Ảnh sẽ được lấy từ các thiết bị cảm biến (camera) và được số hóa nếu thiết bị thu nhận không có chức năng số hóa ảnh đầu vào. Ảnh sẽ được số hóa từ bộ chuyện đổi tín hiệu tương tự qua số.

Nâng cao hình ảnh: Quá trình này xử lý một hình ảnh sao cho kết quả đầu ra nhận được phù hợp hơn so với ảnh gốc ở một ứng dụng cụ thể. Nói cách khác, kỹ thuật này làm rõ các chi tiết ẩn hoặc đơn giản là làm nổi bật một số đặc tính được quan tâm trong một bức ảnh. Đây là một bước quan trọng vì nó thiết lập các thông số, kĩ thuật định hướng theo một vấn đề cụ thể. Ví dụ, một kĩ thuật tăng cường hình ảnh tia X thì nó không phù hợp để sử dụng cho các hình ảnh nhận được từ vệ tinh được chụp từ phổ từ trong dải hồng ngoại. Các phương pháp nâng cao hình ảnh trong quá trình này chỉ mang tính tương đối. Khi một

hình ảnh được xử lý và biểu diễn một cách trực quan, người xem chính là người đánh giá cuối cùng về một phương pháp cụ thể đó có tốt hay không.



Hình 1.4: Các bước cơ bản trong xử lý ảnh số [5].

Khôi phục hình ảnh: Khôi phục hình ảnh là một lĩnh vực liên quan đến việc cải thiện sự xuất hiện của một bức ảnh. Tuy nhiên khác với nâng cao hình ảnh nó có xu hướng là một mô hình toán học hoặc xác suất. Mặt khác việc cải tiến dựa trên sở thích chủ quan của con người thì việc đánh giá tốt hay không cũng trở nên thiếu tính thuyết phục.

Xử lý ảnh màu: Xử lý ảnh màu là bước sử dụng màu sắc của một bức ảnh để lấy ra các đặc tính quan tâm trong một bức ảnh

Phân loại ảnh và xử lý đa biến: Phân loại ảnh và xử lý đa biến là nền tảng trong việc phân chia hình ảnh thành nhiều độ phân giải khác nhau. Nó được sử dụng trong việc nén dữ liệu ảnh.

Nén ảnh: Nén ảnh là kỹ thuật để giảm dung lượng dùng để lưu một bức ảnh hoặc một băng thông cần thiết để truyền đi một bức ảnh. Mặc dù các công nghệ lưu trữ đã được cải thiện đáng kể trong nhiều thập kỷ qua, nhưng điều tương tự thì không thể nói trong khả năng truyền tải ảnh đặc biệt là trong môi trường internet đặc trưng bởi dung lượng của hình ảnh. Nén ảnh rất quen thuộc đối với hầu hết các người dùng máy tính. Chẳng hạn như định dạng ảnh jpg được sử dụng trong tiêu chuẩn nén JPEG (Joint Photographic Experts Group).

Xử lý hình thái học: Xử lý hình thái học sử dụng công cụ dùng để lấy ra các thành phần hình ảnh hữu ích trong việc nhận dạng hay mô tả hình dáng. Trong bước này sẽ có một sự chuyển đổi từ xử lý ảnh đầu ra, để xử lý các đặc tính của ảnh đầu ra.

Phân đoạn ảnh: Phân đoạn ảnh là một bước quan trọng trong một quá trình xử lý ảnh số. Ở bước này các thì ảnh số được phân tách thành các thành phần có cùng một tính chất nào đó dựa theo biên hay các vùng lân cận. Tiêu chuẩn để xác định vùng lân cận có thể là cùng giá trị màu màu, cùng độ sáng… Mục đích của phân đoạn ảnh chính là làm nổi bật hay tách các thuộc tính các thành phần cấu tạo nên ảnh thô. Vì lượng thông tin chứ ảnh rất lớn trong khi các ứng dụng khác nhau chỉ cần lấy thông số của một số đặc trưng cụ thể vì vậy cần có một quá trình để giảm thiểu lượng thông tin khổng lồ. Quá trình này bao gồm phân đoạn ảnh và trích chọn các đặc tính của ảnh.

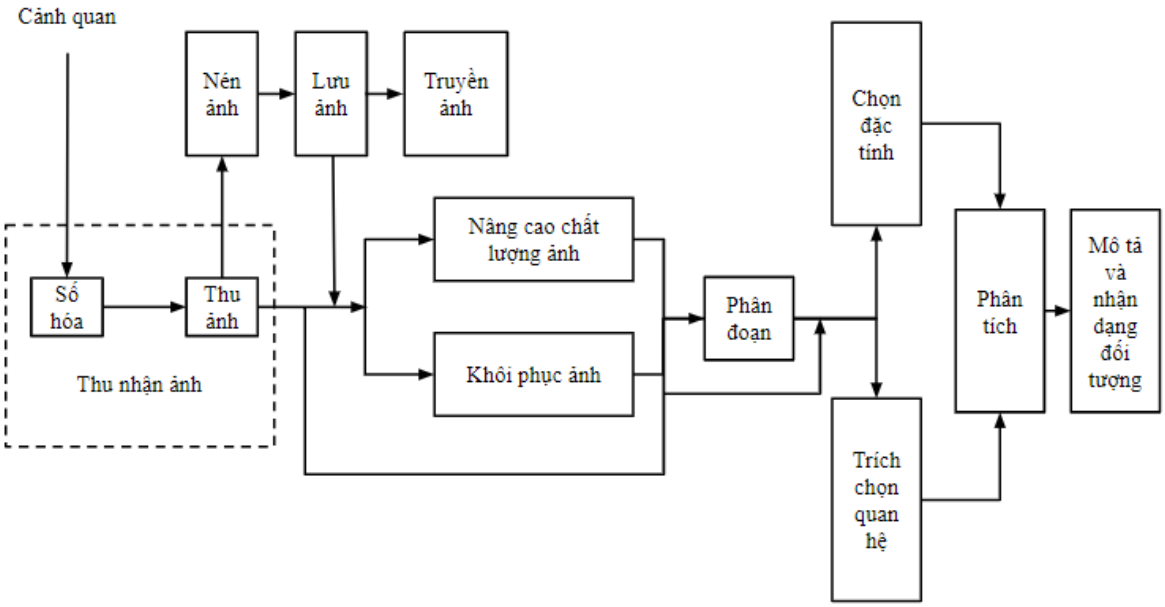
Nhận dạng và mô tả: Kết quả của quá trình phân đoạn ảnh thường được biểu diễn dưới dạng dữ liệu ảnh thô, trong đó có các đặc tính như vùng biên hoặc các đặc tính riêng biệt của chính bức ảnh đó. Để biểu diễn các dữ liệu đó thì câu hỏi đặt ra chính là việc lựa chọn một trong hai phương pháp biến đổi chính là biến đổi dựa trên các vùng biên hay là trích chọn ra các đặc tính của một bức ảnh. Biểu diễn dưới dạng các vùng biên chỉ phù hợp cho các ứng dụng chỉ cần quan tâm đến dáng vẻ bề ngoài mà không cần đi sâu vào phân tích như các góc canh hay hình dáng của một vật. Biểu diễn dưới dạng các đặc tính riêng biệt thì lại thích hợp cho các ứng dụng quan tâm đến các tính chất của đối tượng trong ảnh. Các bước trên chỉ là một phần trong việc biến đổi các dữ liệu của ảnh thô sang một dạng khác phú hợp hơn cho mục đích của người sử dụng. Ngoài ra chúng ta còn cần phải sử dụng các phương pháp khác sao cho làm nổi bật hơn được các đặc tính cần sử dụng cho mục đích sau này.

Nhận dạng đối tượng: Nhận dạng đối tượng có thể hiểu đơn giản là việc gán nhãn cho các đối tượng trong ảnh. Ví dụ đối tượng là con người có xen lẫn ở trong các đối tượng khác thì ta phải tìm cách tách riêng các đặc tính của con người trong ảnh và gãn nhãn để phân biệt với các đối tượng khác.

Cơ sở tri thức: Ảnh là một đối tượng khá phức tạp về đường nét, độ sáng tối, dung lượng điểm ảnh, môi trường để thu ảnh phong phú kéo theo nhiễu [12]. Trong nhiều bước xử lý và phân tích ảnh, ngoài việc đơn giản hóa các đặc tính, tính chất để giảm thiểu việc tính toán cho việc xử lý thì hiện nay việc tiếp cận đến các quy trình để tiếp nhận và xử lý ảnh theo cách của con người cũng mà một mong muốn mà người dùng hướng đến. Cho

nên trong tất các các bước hiện nay đã có áp dụng trí tuệ con người vào xử lý và phát huy một cách hiệu quả.

Trên đây là các thành phần cơ bản trong các bước xử lý ảnh số hoàn chỉnh. Trong thực tế, các quá trình xử lý ảnh số không nhất thiết phải thực hiện hết tất cả các bước mà còn tùy theo đặc điểm ứng dụng. Đối với các ứng dụng chỉnh sửa thì chỉ cần đến các bước tiền xử lý ảnh để nâng cao chất lượng ảnh hiện thị. Còn bước nhận dạng và giải thích đa số chỉ sử dụng cho các ứng dụng phức tạp hay cần xử lý chính xác như các ứng dụng nhận dạng hay phát hiện chuyển động… Hình 1.5 cho sơ đồ phân tích và xử lý ảnh và lưu đồ thông tin giữa các khối một cách khá đầy đủ. Khi tiếp nhập hình ảnh từ các camera hay cảm biển thì đầu tiên ảnh sẽ được số hóa, nén lại để giảm thiểu dung lượng và lưu vào để sẵn sàng cho việc sử dụng trực tiếp hoặc để xử lý các bước tiếp theo. Tuy nhiên, ảnh sau khi được số hóa có thể bỏ qua các bước tiền xử lý ảnh như nâng cao chất lượng ảnh hay khôi phục ảnh để chuyển đến bước phân đoạn hoặc bỏ tiếp bước phân đoạn chuyển trực tiếp tới bước trích chọn các đặc trưng, quan hệ trong bức ảnh. Sơ đồ biểu diễn cụ thể quan hệ của các bước trong xử lý ảnh số, từ đó có thể giảm thiểu hay sử dụng các bước này một cách hiệu quả nhất đối với mục đích của người sử dụng.



Hình 1.5: Mối liên hệ giữa các bước trong xử lý ảnh số [12].

1.3 Một số vấn đề trong xử lý ảnh số

Để xử lý một bức ảnh số thì có rất nhiều kĩ thuật thực hiện cùng với các mục đích khác nhau. Sau đây sẽ là một số vấn đề cơ bản trong xử lý ảnh số.

1.3.1 Chỉnh mức xám

Nhằm khắc phục tính không đồng đều trong một bức ảnh thì có hai bước để tiếp cận:

Giảm số mức xám: Thực hiện bằng cách nhóm các mức xám với giá trị xấp xỉ nhau thành một nhóm. Trong trường hợp chỉ có hai mức xám thì bức ảnh đầu ra chính là ảnh đen trắng. Bước này được dùng trong các máy in ảnh màu ra đen trắng.

Tăng số mức xám: Bước này thực hiện thêm các mức xám trung gian bằng kỹ thuật nội suy. Ảnh khi tăng mức xám sẽ có thêm các giá trị về độ sáng từ đó ảnh sẽ mịn hơn và tiếp cận tới mắt người dùng một cách chân thật hơn.

1.3.2 Biến đổi ảnh

Trong xử lý ảnh, khi số điểm ảnh càng lớn thì việc tính toán cũng có độ phức tạp ngày càng cao từ đó đòi hỏi về dung lượng lưu trữ cũng như thời gian thực hiện lơn hơn rất nhiều. Các phương pháp khoa học kinh điển áp dụng cho xử lý ảnh hầu hết khó khả thi. Người ta sử dụng các phép toán tương đương hoặc biến đổi sang miền xử lý khác để dễ tính toán. Sau khi xử lý dễ dàng hơn được thực hiện, dùng biến đổi ngược để đưa về miền xác định ban đầu, các biến đổi thường gặp trong xử lý ảnh gồm [12]:

- Biến đổi Fourier, Cosin, Sin
- Biến đổi (mô tả) ảnh bằng tích chập, tích Kronecker
- Các biến đổi khác như KL (Karhumen Loeve), Hadamard

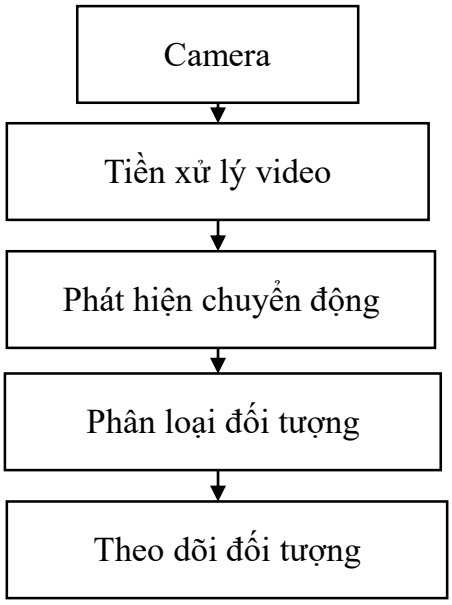
1.3.3 Nén ảnh

Một bức ảnh dù được biến đổi ở dạng nào cũng cần một không gian rất lớn để lưu trữ. Khi đó các phương pháp nén ảnh được đưa vào để giảm thiểu không gian bộ nhớ. Hiện nay, các chuẩn nén ảnh gồm MPEG, H264, H265 đang được sử dụng rộng rãi và phát huy một cách hiệu quả.

1.4 Thuật toán theo dõi và phát hiện chuyển động.

Theo dõi chuyển động của con người là một vấn đề phức tạp vì hình dạng cơ thể, khớp nối, tốc độ chuyển động nhanh và quần áo. Ngoài ra còn có thêm nhiều vấn đề phức

tạp khác như điều kiện ánh sáng, ảnh nền và các nhiễu có thể ảnh hưởng rất lớn trong một số kỹ thuật. Để đạt được mục đích cuối cùng thì phải cần có bước tiền xử lý video trước khi tiến hành các bước khác để thực hiện các bước khác. Sau khi xử lý được các vấn đề ngoại cảnh thì chúng ta cần phát hiện và phân loại đối tượng trước khi có thể theo dõi nó. Các kỹ thuật khoanh vùng đối tượng để phát hiện đối tượng quan tâm trong vùng video và phân cụm các pixel tương ứng với vị trí các đối tượng này. Các đối tượng được phát hiện chuyển động có thể nhiều loại như con người, động vật hoặc có thể như cây cối đung đưa theo gió… Sau đó chúng ta có thể theo dõi đối tượng dễ dàng hơn. Các bước theo dõi đối tượng được thể hiện như Hình 1.6:



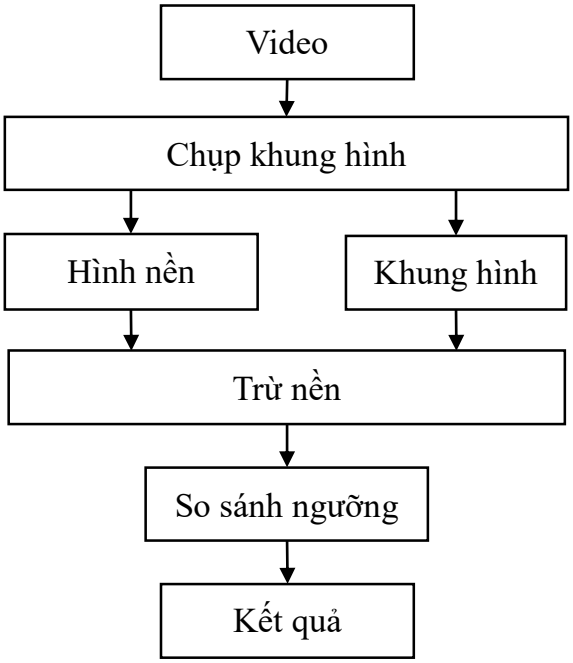
Hình 1.6: Các bước theo dõi đối tượng.

Trước khi tiến hành việc theo dõi đối tượng chúng ta phải phát hiện được các đối tượng trong các khung hình video. Vì vậy ở phần tiếp theo học viên sẽ trình bày hai thuật toán phát hiện đối tượng cơ bản là thuật toán trừ nền và thuật toán chênh lệch khung hình.

1.4.1 Thuật toán trừ nền

Thuật toán trừ nền (Background Subtraction Algorithms) là cách tiếp cận phổ biến nhất để xác định đối tượng chuyển động là thực hiện thuật toán trừ nền, trong đó mỗi pixel từ khung hình của video hiện tại được so sánh với hình ảnh tham chiếu chính là hình ảnh nền. Khi các pixel trong khung hình hiện tại khác biết một giá trị đáng kể so với pixel ở khung hình nên thì đối tượng được coi là chuyển động. Nó cũng tương tự thuật toán được diễn tả ở phần trên khi lấy các giá trị sai khác của các pixel tại một thời điểm xem xét. Tuy

nhiên nên ở thuật toán này được xác định là nền tĩnh và không thay đổi luôn luôn cố định khiến cho kết quả đầu ra được chính xác hơn. Các bước thực hiện thuật toán được biểu diễn trong sơ đồ dưới đây:



Hình 1.7: Sơ đồ khối thuật toán trừ nền.

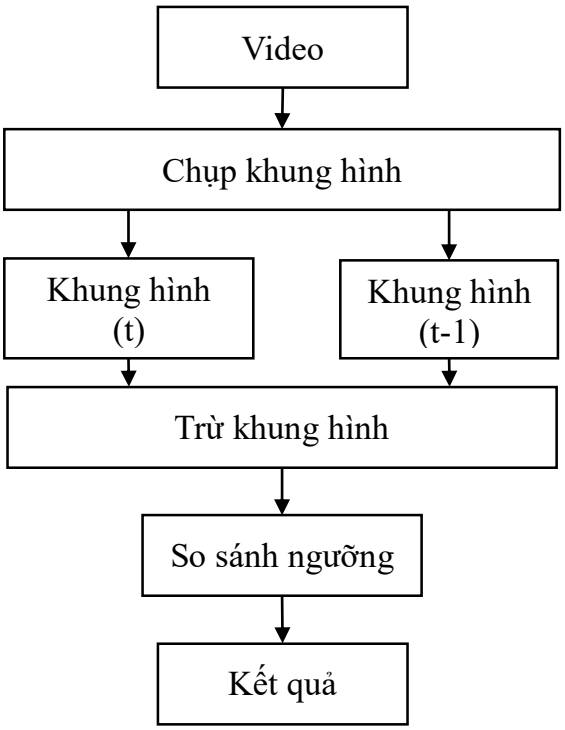
Thuật toán này cũng rất đơn giản và có được kết quả cực kì chính xác bởi vì nên là cảnh tĩnh là không thay đổi hệ quy chiếu trong suốt quá trình xử lý. Tuy nhiên cũng có điểm hạn chế từ việc thực hiện nền tĩnh đó chính là việc thay đổi các yếu tố môi trường như thay đổi cường độ sáng, vị trí camera bị xê dịch hay hình ảnh nền có nhiều vị trí không phải là vật tĩnh…

1.4.2 Thuật toán trừ khung hình

Thuật toán trừ khung hình (Frame Difference Algorithm) bắt đầu hoạt động với giả định nền là cảnh tĩnh và so sánh các pixel của các khung hình được chụp trong một khoảng thời gian rất nhỏ là (Δt). Nếu có sự khác biệt với giá trị trên một ngưỡng (Γ), điều đó chứng tỏ các pixel tại địa chỉ khung hình đó đã bị thay đổi và có thể kết luận rằng có chuyển động. Thuật toán được biểu diễn bởi công thức sau:

$$I_i = \begin{cases} 1 & \text{nếu } |f_t - f_{t-1}| \geq \Gamma \\ 0 & \text{nếu } |f_t - f_{t-1}| < \Gamma \end{cases} \quad (1.1)$$

Trong đó I_i là sự sai của một pixel giữa hai khung hình video liên tiếp là f_t và f_{t-1}. Sự khác biệt của tất cả các pixel tại một thời điểm của một khung hình sẽ là một tập hợp số nhị phân từ đó sẽ cho chúng ta thấy được sự chuyển động so với khung hình trước đó. Thuật toán được thể hiện trong sơ đồ khối sau:



Hình 1.8: Sơ đồ khối thuật toán trừ khung hình.

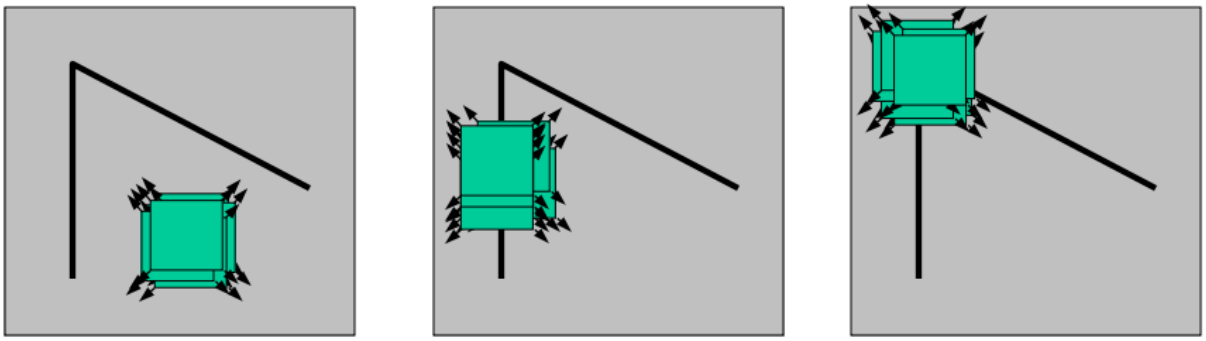
Phương pháp này có ưu điểm là cho phép nhận dạng chuyển động tốt, dễ thực thi, không cần đòi hỏi nhiều không gian bộ nhớ và có thể hoạt động được khi camera bị xê dịch hay có các yếu tố môi trường. Tuy nhiên có một hạn chế chính là thuật toán có không phải là nền tĩnh cho nên việc chuyện động đôi khi là tương đối bởi vì hệ quy chiếu nên luôn luôn thay đổi theo thời gian.

1.5 Thuật toán phân loại đối tượng

Các phương pháp phát hiện khoanh vùng chuyển động đã được đề cập ở phần trước. Do đó chúng ta cần sử dụng các phương pháp khác để có xác định được đối tượng mà chúng ta cần theo dõi trong các đối tượng chuyển động. Việc này đòi hỏi một quá trình rất phức tạp và khó khăn. Đối tượng được theo dõi luôn chuyển động và có thể có vật cản hay góc khuất chắn đi góc nhìn tham chiều của camera, ngoài ra còn có yếu tố môi trường được đề cập ở phần trước cũng ảnh hưởng một phần không nhỏ đến việc theo dõi đối tượng được

chỉ định. Cho nên chúng ta cần phải tập trung đến các đặc tính nổi bật của đối tượng. Màu sắc, hình dạng, kết cấu và thậm chí các thuộc tính của chuyển động của đối tượng có thể giúp nhận dạng nó. Sau đây học viên sẽ giới thiệu về thuật toán theo dõi nhận dạng đối tượng.

Thuật toán Harris Corner Detector được đề xuất bởi Harris và Stephens vào năm 1988, nó là phương pháp phát hiện góc (conner detector). Góc ở trong thuật toán này được sử dụng trong hình ảnh 2D có hai cạnh và biên có hướng khác nhau trong các điểm lân cận. Từ một vùng cửa sổ sau đó dịch dần vào vùng trung tâm mà chung ta cần xác định đối tượng, góc chính là các điểm có sự thay đổi lớn về cường độ các pixel trong điểm đó và nó được thể hiện rõ trong hình 1.9 vẽ dưới đây:



Vùng trống Biên Góc

Hình 1.9: Cách nhận biết góc trong thuật toán Harris Conner

Các góc được xem như là điểm quan tâm bởi vì chúng cho chúng ta các giá trị đúng và khác nhau trong khi vùng trống và các biên thì cho quá nhiều các điểm ảnh khác nhau có kết quả tương ứng. Để tìm những sự thay đổi khác nhau này thuật toán này dựa trên một hàm tính toán tự động như sau:

$$E(u,v) = \sum_{x,y} w(x,y)^2 (I(x+u, y+v) - I(x,y))^2 \quad (1.2)$$

Trong đó w(x,y) chính là hàm cửa số, I(x,y) chính là cường độ sáng của điểm ảnh tại tọa độ (x,y) và $I(x+u, y+v)$ chính là cường độ sáng sau khi dịch chuyển cửa sổ đến điểm (x+u, y+v). Chúng ta có thể dùng bộ lọc trong đó các điểm ảnh trong vùng lân cận có giá trị bằng 1 và tất cả các điểm ảnh còn lại có giá trị là 0. Hoặc chúng ta cũng có thể áp dụng bộ lọc Gaussian trong đó các điểm ảnh lân cận, các điểm ảnh đang tính toán có giá trị cao

hơn so với các điểm còn lại theo phân bố Gaussian. Áp dụng biến đổi Taylor cho các cường độ thay đổi chúng ta có:

$$I(x+u, y+v) \approx I(x,y) + uI_x + vI_y \tag{1.3}$$

Với I_x và I_y là các đạo hàm riêng tương ứng theo x và y. Biến đổi ra chúng ra được hàm:

$$E(u,v) \approx \sum_{x,y} w(x,y)^2 \left(uI_x + vI_y\right)^2 = \sum_{x,y} w(x,y)^2 \left([u \quad v] \begin{bmatrix} I_x \\ I_y \end{bmatrix}\right)^2 \tag{1.4}$$

Sử dụng các phép nhân tính toán ma trận chúng ta được kết quả:

$$E(u,v) \approx [u \quad v]\, S_a \begin{bmatrix} u \\ v \end{bmatrix} \tag{1.5}$$

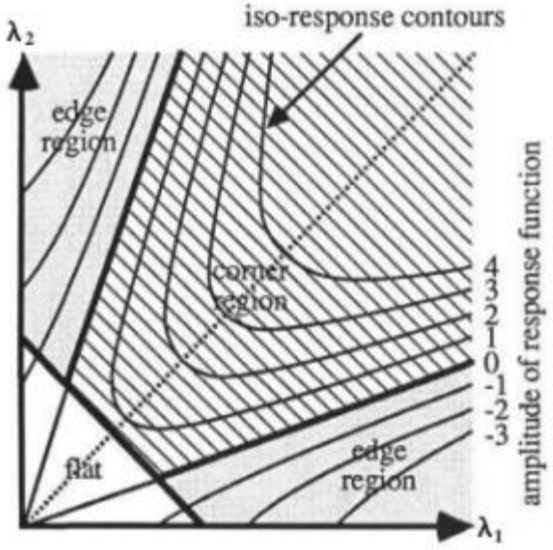
Trong đó $S_a = \sum_{x,y} w(x,y)^2 \begin{bmatrix} I_x I_x & I_x I_y \\ I_y I_x & I_y I_y \end{bmatrix}$ với giá trị riêng của ma trận S_a là λ_1 và λ_2, k là một hằng số và được xác định thông qua thực nghiệm, thường có giá trị trong khoảng [0,04….0,15]. Khi đó biểu thức dưới đâu sẽ quyết định hàm cửa sổ w có chưa góc hay không:

$$M_c = \lambda_1\lambda_2 - k(\lambda_1 + \lambda_2)^2 = \det(S_a) - k(\mathrm{trade}^2(S_a)) \tag{1.6}$$

Nếu giá trị cả λ_1 và λ_2 đều nhỏ. Có nghĩa là hàm của sổ E(u,v) gần như không thay đổi theo bất kì hướng nào. Khi đó vùng cửa sổ các vùng lân cận hâu như không có sự thay đổi về cường độ sáng. Điều này cho thấy cửa sổ không tìm ra góc hoặc cạnh nào.

Nếu giá trị λ_1 lớn và λ_2 nhỏ, hoặc ngược lại. Thì có nghĩa là hàm cửa sổ E(u,v) có sự thay đổi về giá trị khi trượt về một hướng, và có sự thay đổi đáng kể nếu dịch chuyển theo hướng trực giao. Điều này cho thấy tồn tại một cạnh.

Nếu giá trị cả λ_1 và λ_2 đều lớn. Thì khi đó sự dịch chuyển của cửa sổ đến một hướng bất kì đều có sự thay đổi cường độ sáng. Điều này biểu thị cho một điểm góc xuất hiện. Biểu đồ về quan hệ của λ_1 và λ_2 được thể hiện trong Hình 1.10 [11] dưới đây:

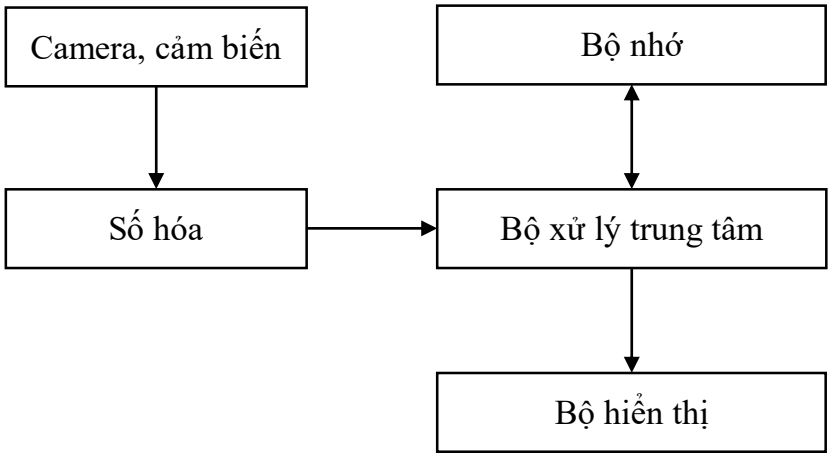


Hình 1.10: Biểu đồ quan hệ giữa λ_1 và λ_2 [11].

Như vậy chúng ta đã thực hiện được các bước xử lý ảnh từ phát hiện đối tượng chuyển động và nhận dạng đối tượng để theo dõi bằng các thuật toán khác nha. Tiếp theo sẽ là một số giải pháp phần cứng để có thể thực hiện thuật toán xử lý ảnh và sau đó chọn ra một giải pháp phù hợp để sử dụng.

1.6 Hệ thống xử lý ảnh số

Hệ thống xử lý trên máy tính số là một hệ thống gồm bộ thu nhận ảnh (máy ảnh, cảm biến), bộ số hóa, bộ xử lý trung tâm, màn hình và bộ nhớ như hình 1.11. Hiện nay phần lớn các hệ thống xử lý ảnh đều đưa về ảnh xám hoặc ảnh đen trắng để xử lý mục đích là giảm độ phức tạp cũng như khối lượng tính toán cho hệ thống. Ảnh xám được áp dụng trong rất nhiều lĩnh vực đời sống như sinh vật học, y học hoặc trong công nghiệp. Thực tế chỉ ra rằng bất kỳ ứng dụng nào xử lý được trên ảnh xám cũng có thể ứng dụng được trên ảnh màu. Tuy nhiên, với sự phát triển trong công nghiệp điện tử trong việc tích hợp nhiều thành phần trong một hệ thống thì bây giờ chúng ta có thể thực hiện đồng thời biểu diễn cả hai trên một hệ thống mà vẫn đảm bảo được hiệu năng cũng nhưng dung lượng lưu trữ của hệ thống. Vì vậy, các bức ảnh được xử lý ngày càng chính xác và tốc độ xử lý cũng ngày càng nhanh để ứng dụng vào nhiều lĩnh vực cần độ chính xác và khả năng làm việc trên thời gian thực cao.



Hình 1.11: Hệ thống xử lý ảnh.

1.7 Một số giải pháp phần cứng thực thi trong hệ thống xử lý ảnh

Trong một hệ thống xử lý ảnh thì các thuật toán về xử lý ảnh tương đối phức tạp và khối tượng tính toán cũng cực kì lớn. Cho nên để thỏa mãn yêu cầu về tốc độ và hiệu suất xử lý thì đòi hỏi hệ thống phần cứng được sử dụng phải đủ mạnh và khả năng trễ thấp để có thể làm việc trên thời gian thực.

Dưới đây là một số giải pháp phần cứng hiện nay được sử dụng cho một hệ thống xử lý ảnh.

Field Programmable Gate Array (FPGA)

FPGA là một mạch tích hợp cỡ lớn dùng cấu trúc mảng phần tử lô-gic mà người dùng có thể lập trình và thay đổi thiết kế của mình. Đây là một giải pháp phần cứng hay được sử dụng hiện nay tận dụng các đặc tính của FPGA là tính linh hoạt và tốc độ xử lý nhanh đặc biệt cùng khả năng xử lý song song để tăng tốc độ xử lý thì FPGA rất phù hợp để xử lý các thuật toán xử lý ảnh với mức độ tính toán phực tạp và dung lượng lớn.

Digital Signal Processor (DSP)

DSP được giới thiệu đầu tiên vào năm 1978, 1979 bởi Intel, Bell Labs. Nó là một bộ vi xử lý có thể lập trình cho một mục đích chuyên dụng nào đó, được thiết kế để điều khiển theo thời gian thực với luồng truyền liên tục của một khối dữ liệu số nhằm cải thiện chất lượng hay sử đổi bổ sung theo yêu cầu riêng. DSP có các đặc điểm sau:

Thích hợp cho các quá trình xử lý theo thời gian thực
Hiệu năng được tối ưu hóa với dữ liệu dạng luồng

Chương trình và dữ liệu được phân bố riêng biệt
Tích hợp các chỉ thị lệnh đặc biệt
Tương tác trực tiếp với bộ nhớ của thiết bị
Tích hợp sẵn DAC và ADC

DSP hiện nay còn tích hợp thêm nhiều thành phần khác nhau để làm tăng tính linh hoạt và tốc độ xử lý. Cho nên DSP thích hợp cho những nhu cầu tính toán nhanh, xử lý số thực. Với các ưu điểm đó thì DSP là một giải pháp thích hợp cho một hệ thống xử lý ảnh.

Mainboard, labtop

Đây là giải pháp đơn giản nhất khi có thể sử dụng máy tính xách tay như một hệ thống xử lý ảnh và đưa ra các quyết định. Với việc tích hợp các camera hay webcam một cách dễ dàng và các giao tiếp ngoại vi phổ biến như keyboard, USB, UART. Ngoài ra nó sẽ có một số ưu điểm như:

Tốc độ xử lý cao
Dễ dàng lập trình kiểm lỗi
Hệ điều hành dễ sử dụng
Tích hợp nhiều công cụ lập trình phổ biến

Tuy nhiên đi cùng với đó là các nhược điểm như hệ thống phần cứng không thể can thiệp chỉ thao tác được trên phần mềm. Có rất nhiều thành phần không cần thiết trong mục đích xử lý ảnh gây sự lãng phí không cần thiết.

Đề tài này đề cấp đến một hệ thống xử lý ảnh với mục đích có thể hiệu chỉnh được hệ thống và đảm bảo tính ổn định cũng như tốc độ xử lý và hiệu năng cao. Thì trong các giải pháp đã đề cập ở trên thì việc thực thi hệ thống xử lý ảnh bằng công nghệ FPGA chính là giải pháp thích hợp nhất.

Chương 2

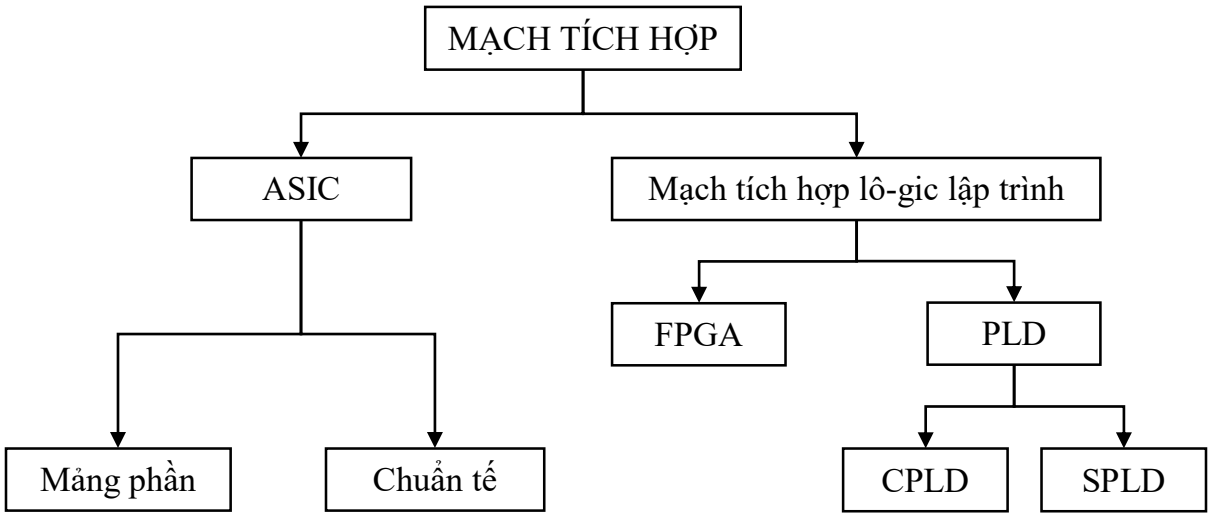
CÔNG NGHỆ FPGA

Field Programmable Gate Array (FPGA) đã trở thành một lĩnh vực quan trọng trong thiết kế và thực thi mạch số trong thập kỉ qua. Kiến trúc FPGA có ảnh hưởng rất lớn đến hiệu năng, hiệu quả khu vực và mức tiêu thụ năng lượng của thiết bị. Vì vậy, muốn cải tiến một thiết bị chúng ta phải cái tiến về phần cứng chứ không chi riêng phần mềm và chương này sẽ giới thiệu sơ qua tổng quát về công nghệ FPGA.

2.1 Tổng quan về mạch tích hợp

Một mạch tích hợp IC (thường được gọi là vi mạch) là tập các mạch điện chứa các linh kiện bán dẫn và linh kiện điện tử thụ động (điện trở) được kết nối với nhau, để thực hiện một chức năng nhất định. Các mạch tích hợp với các linh kiện có kích thước cỡ micrometre hoặc nhỏ hơn được chế tạo bằng công nghệ silicon, mỗi một tấm silicon chứa hàng trăm vi mạch nhỏ. Mạch tích hợp giúp giảm kích thước của mạch điện đi rất nhiều, bên cạnh đó là độ chính xác tăng lên. Mạch tích hợp là một phần rất quan trọng của các mạch lô-gic. Có hai loại mạch tích hợp chính gồm lập trình được và không lập trình được được mô tả trong hình 2.1. Hiện nay, công nghệ silicon đang tiến tới những giới hạn của vi mạch tích hợp và các nhà nghiên cứu đang nỗ lực tìm ra một loại vật liệu mới có thể thay thế công nghệ silicon này.

Lịch sử phát triển của mạch tích hợp bắt đầu từ năm 1949, khi kỹ sư người Đức Werner Jacobi (Siemens AG) nộp bằng sáng chế cho một thiết bị khuếch đại bán dẫn giống như mạch tích hợp, có 5 transistor trên một bề mặt chung cho bộ khuếch đại 3 tầng, làm dụng cụ trợ thính. Mạch tích hợp đầu tiên do Jack Kilby (Texas Instrumens) phát minh ngày 12/09/1958.



Hình 2.1: Các loại mạch tích hợp.

Ngày nay người ta phân loại các mạch tích hợp dựa theo tiêu chí về tỉ lệ tích hợp linh kiện bán dẫn trong một mạch tích hợp như sau:

Mạch tích hợp cỡ nhỏ (SSI): có tới 100 transistors trong chip

Mạch tích hợp vừa (MSI): từ 100 đến 3.000 transistors trong chip

Mạch tích hợp cỡ lớn (LSI): từ 3.000 đến 100.000 transistors trong chip (1970)

Mạch tích hợp cỡ rất lớn (VLSI): từ 100.000 đến 1.000.000 transistors trong chip (1980)

Mạch tích hợp cỡ siêu lớn ULSI): từ vài triệu đến vài tỷ transistors trong chip.

Các phần sau đề cập đến một số loại mạch tích hợp điển hình hiện nay.

2.1.1 Mạch tích hợp ứng dụng chuyên biệt

Mạch tích hợp ứng dụng chuyên biệt (ASIC) là một mạch tích hợp được thiết kế dành cho các ứng dụng cụ thể. Mạch tích hợp ứng dụng chuyên biệt ngày nay được ứng dụng hầu như khắp mọi nơi, ví dụ như vi xử lý của điện thoại di động, vi xử lý trong các máy móc tự động, các phương tiện truyền thông, xe cộ, tàu vũ trụ, các hệ thống xử lý, các dây chuyền công nghiệp … Ngày nay, với sự phát triển mạnh mẽ của các công cụ thiết kế, các chức năng của một sản phẩm ASIC ngày càng nhiều, số lượng cổng có thể lên đến hàng trăm tỉ cổng. Tuy nhiên, đối với nhiều cá nhân, phòng thì nhiệm thì chi phí để trả cho thiết kế chế tạo (tại nhà máy) các mạch tích hợp ứng dụng chuyên biệt là rất cao. Nên một số nhà cung cấp đã tạo ra các mạch tích hợp có chi phí thấp hơn, đó là các mạch tích hợp mà

người sử dụng có thể tự thiết kế và lập trình các vi mạch để tạo ra các thiết bị phù hợp nhất cho nhu cầu chuyên dụng của mình.

Tiếp theo, học viên sẽ đi phân tích hai loại mạch tích hợp có thể cấu hình được là mảng cổng lô-gic có khả năng lập trình và các thiết bị lô-gic có thể lập trình được.

2.1.2 Các thiết bị lô-gic có thể lập trình được

Các thiết bị lô-gic có thể lập trình được (PLD) là một mạch tích hợp số được sử dụng để chế tạo mạch kĩ thuật số có thể cấu hình lại, chúng có khả năng thực hiện các chức năng lô-gic như mong muốn. Đây là một chip loại mạch tích hợp cỡ lớn có chứa một cấu trúc cho phép tùy biến tức là có thể lập trình cấu hình lại để thực hiện một chức năng cần thiết cho ứng dụng của họ. Tuy nhiên, các thiết bị này có thể cấu hình hoặc được lập trình để tạo một phần cho ứng dụng chuyên biệt, và như vậy chúng có thể được coi là thuộc họ các mạch tích hợp ứng dụng chuyên biệt. Nhưng các thiết bị lô-gic có thể lập trình được sử dụng các công nghệ khác so với mạch tích hợp ứng dụng chuyên biệt để lập trình. Có hai loại thiết bị lô-gic có thể lập trình được: thiết lô-gic khả trình đơn giản (SPLD), thiết bị lô-gic khả trình phức tạp (CPLD).

Thiết bị lô-gic khả trình phức tạp được cấu trúc từ số lượng nhất định các thiết bị lô-gic khả trình đơn giản. Các khối thiết bị lô-gic khả trình đơn giản thường là một mảng lô-gic AND/OR lập trình được có kích thước xác định và chứa một số lượng hạn chế các phần tử nhớ đồng bộ. Cấu trúc này hạn chế khả năng thực hiện những hàm phức tạp và thông thường hiệu suất làm việc của vi mạch phụ thuộc vào cấu trúc cụ thể của vi mạch hơn là vào yêu cầu bài toán.

2.1.3 Mảng cổng lô-gic có khả năng lập trình

Mảng cổng lô-gic có khả năng lập trình (FPGA) là một mạch tích hợp cỡ lớn dùng cấu trúc mảng phần tử lô-gic mà người dùng có thể lập trình và thay đổi thiết kế của mình. Mảng cổng lô-gic có khả năng lập trình có khả năng tái lập trình bởi người sử dụng mà không phụ thuộc vào dây truyền sản xuất phức tạp của nhà máy bán dẫn.

Mảng cổng lô-gic có khả năng lập trình được xem như một mạch tích hợp ứng dụng chuyên biệt, nhưng nếu so sánh thì nó không đạt được mức độ tối ưu như mạch tích hợp này như hạn chế khả năng thực hiện được những tác vụ đặc biệt phức tạp. Tuy vậy, các công nghệ ASIC thường mất hàng tháng để chế tạo và chi phí tiêu tốn rất cao để cho ra sản phẩm đầu tiên. Mảng cổng lô-gic có khả năng lập trình có khả năng tái cấu hình mạnh,

công đoạn thiết kế đơn giản, do vậy, việc sử dụng công nghệ FPGA trong quá trình thiết kế trước khi đưa ra sản xuất các mạch ASIC mẫu đóng một vai trò vô cùng to lớn trong việc giảm giá thành và thời gian chế tạo các mạch tích hợp này.

Bản chất linh hoạt của công nghệ FPGA còn được kể đến diện tích, độ trễ và mức tiêu thụ điện năng. Một vi mạch FPGA có diện tích lớn hơn khoảng 20 đến 35 lần so với vi mạch ASIC, tốc độ hiệu năng kém ba đến bốn lần và tiêu thụ năng lượng động gấp 10 lần. Những nhược điểm này phát sinh phần lớn từ khối lập trình định tuyến của FPGA, nơi phụ trách về việc thực thi, kết nối, và tiêu thụ năng lượng để đổi lấy việc chế tạo tức thời.

Thiết kế hay lập trình cho mạch FPGA được thực hiện chủ yếu bằng các ngôn ngữ mô tả phần cứng như VHDL, verilog …

2.2 So sánh mảng cổng lô-gic có khả năng lập trình và các mạch tích hợp khác

2.2.1 Mảng cổng lô-gic có khả năng lập trình và mạch tích hợp ứng dụng chuyên biệt

Các mạch tích hợp ứng dụng chuyên biệt có hiệu năng, dụng lượng lưu trữ cao hơn và lại có nguồn nuôi thấp hơn so với mảng cổng lô-gic có khả năng lập trình. Các thiết kế mảng cổng lô-gic có khả năng lập trình tiêu thụ năng lượng lớn hơn so với mạch tích hợp ứng dụng chuyên biệt. Do vậy nếu để sản xuất công nghiệp với số lượng lớn thì công nghệ ASIC cho chi phí sản xuất trên một đơn vị thấp hơn các sản phẩm từ công nghệ FPGA.

Mảng cổng lô-gic có khả năng lập trình phù hợp sử dụng cho các ứng dụng vừa và nhỏ bởi vì nó có chi phí thiết kế thấp. Nhưng nếu đưa vào sản xuất công nghiệp với số lượng lớn, thì mạch tích hợp ứng dụng chuyên biệt lại có giá rẻ hơn nhiều bởi công nghệ ASIC đi theo hướng thiết kế cho một ứng dụng cụ thế chứ không đi theo hương linh hoạt và tái cấu hình như công nghệ FPGA.

Quá trình thiết kế mảng cổng lô-gic có khả năng lập trình đơn giản và thời gian ngắn hơn so với quá trình thiết kế mạch tích hợp ứng dụng chuyên biệt, bởi vì không cần phải sắp xếp linh kiện, không cần các mặt mạ hoặc các quá trình sau cuối.

Mạch tích hợp ứng dụng chuyên biệt có thể có các thiết kế xử lý hỗn hợp các tín hiệu, hoặc chỉ là các thiết kế tương tự. Nhưng không thể thiết kế mạch này sử dụng các chip FPGA.

Mạch tích hợp ứng dụng chuyên biệt có thể có các thiết kế hoàn toàn cho các ứng dụng riêng và phức tạp, ví dụ vi xử lý hay bộ nhớ, còn mảng cổng lô-gic lập trình được thì không thể.

Bởi vì mảng cổng lô-gic lập trình được có thể lập trình cấu hình lại vô số lần nên nó phù hợp để giảm thiểu chi phí chế tạo sản phẩm sau cùng. Như vậy với mảng cổng lô-gic lập trình được ta có thể tự thiết kế cả một CPU theo mong muốn.

Các thiết kế mạch tích hợp ứng dụng chuyên biệt phải tốn chi phí rất lớn cho một lần nghiên cứu, thiết kế, và kiểm thử sản phẩm mới, trong khi đó thì các thiết kế dựa trên công nghệ FPGA lại không cần vì khả năng tái cấu hình. Quá trình thiết kế mạch tích hợp ứng dụng chuyên biệt sẽ kéo dài hơn so với mảng cổng lô-gic lập trình được.

Các công cụ được sử dụng cho thiết kế FPGA thường rẻ hơn so với các công cụ thiết kế của ASIC. Một mạch FPGA có thể được sử dụng cho các ứng dụng khác nhau, nhờ khả năng lập trình lại nhưng với ASIC thì không thể. Đây là lý do quan trọng nhất khiên cho công nghệ FPGA ngày càng được sử dụng rộng rãi.

2.2.2 Mảng cổng lô-gic lập trình được và thiết bị lô-gic lập trình được

Để so sánh với thiết bị lô-gic lập trình được thì mảng cổng lô-gic lập trình được tối ưu hơn ở các điểm:

- Công cụ lập trình của FPGA thực hiện đơn giản hơn.
- Kiến trúc của FPGA là kiến trúc mảng các khối lô-gic, các khối lô-gic này nhỏ hơn nhiều nếu đem so sánh với một khối thiết bị lô-gic khả trình đơn giản, ưu điểm này giúp FPGA có thể chứa nhiều hơn các phần tử lô-gic (FPGA có khả năng chứa tới từ 100.000 đến hàng vài tỷ cổng lô-gic, trong khi thiết bị lô-gic khả trình phức tạp chỉ chứa từ 10.000 đến 100.000 cổng lô-gic. Để đạt được mục đích này thì kiến trúc của một mảng cổng lô-gic lập trình được phức tạp hơn nhiều so với một thiết bị lô-gic khả trình phức tạp.
- Các trễ trong các thiết bị lô-gic khả trình phức tạp lớn hơn so với các trễ trong các mảng cổng lô-gic lập trình được.

Ngoài ra mảng cổng lô-gic lập trình được và thiết bị lô-gic lập trình được còn có một số đặc điểm khác nhau như:

Các FPGA gồm có các khối lô-gic nhỏ, trong khi các CPLD được làm ra từ các khối lô-gic lớn.

FPGA là chip lô-gic số dựa vào RAM, trong khi CPLD là chip dựa vào EEPROM.

Các FPGA dựa vào các bảng ánh xạ bên trong, trong khi các CPLD lại hình thành các hàm lô-gic nhờ các mạch cổng sea-of-gates.

Hầu hết các FPGA có các mạch lô-gic mức cao, ví dụ, các bộ cộng, bộ nhân và các bộ nhớ nhúng, và các khối lô-gic thực hiện các bộ giải mã hoặc các hàm toán học, trong khi đó CLPD thì không.

Các mảng cổng lô-gic lập trình được phù hợp cho các thiết kế phức tạp hơn so với thiết bị lô-gic khả trình phức tạp. Nhìn chung một thiết bị lô-gic khả trình phức tạp là sự lựa chọn tốt cho các ứng dụng tổ hợp, trong khi các mảng cổng lô-gic lập trình được lại phù hợp hơn cho các máy trạng thái lớn (vi xử lý). Mảng cổng lô-gic lập trình được có các phần tử lô-gic chạy theo dạng song song. Còn vi điều khiển dựa trên cấu trúc CPU thực thi theo mã lệnh theo dạng tuần tự.

Một điểm khác biệt nữa so với thiết bị lô-gic khả trình phức tạp là trong những mảng cổng lô-gic lập trình được hiện đại được tích hợp nhiều những bộ lô-gic số học đã sơ bộ tối ưu hóa, hỗ trợ RAM, ROM tốc độ cao, hay các bộ nhân cộng.

Sau khi so sánh và nêu ra các ưu nhược điểm của FPGA so với các mạch tích hợp khác học viên nhận thấy sử dụng FPGA cho đề tài lần này là khả thi và phù hợp nhất (chi phí, công cụ, thời gian chế tạo, hiệu năng hoạt động) so với hai mạch tích hợp còn lại. Tiếp theo học viên sẽ đi vào phân tích cơ bản về kiến trúc FPGA và một số ứng dụng của nó.

2.3 Cấu trúc mảng cổng lô-gic lập trình được

Có hai phương pháp lập trình FPGA: lập trình dựa trên SRAM và lập trình dựa trên kết nối cầu chì.

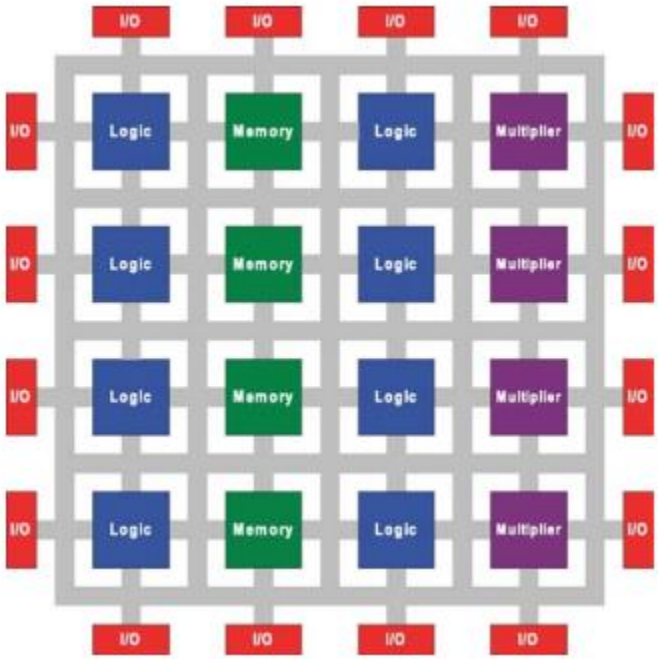
Lập trình dựa trên SRAM cần sử dụng các bit của SRAM cho từng phần tử lập trình. Với giá trị bit là 0 thì tắt chuyển mạch, với giá trị bit là 1 thì bật chuyển mạch. Đối với phương pháp thứ hai, khi lập trình dòng lập trình tạo ra kết nối cầu chì.

FPGA dựa trên SRAM do các SRAM của FPGA có thể được ghi đọc như SRAM bình thường ngay cả khi chúng ở trong hệ thống, nên các FPGA có thể được lập trình lại nhiều lần. Tuy nhiên với việc lập trình FPGA loại này thì trễ định tuyến lại lớn. Loại FPGA này thường được sử dụng cho lập trình cấu hình các mẫu thử của các thiết kế phần cứng trên ASIC.

FPGA dựa vào kết nối cầu chì duy trì cố định nội dung lập trình ngay cả khi mất nguồn, và trễ định tuyến nhỏ. Tuy nhiên chúng yêu cầu một quá trình sản xuất phức tạp,

và nếu đã lập trình xong một lần thì không thể thay đổi được nữa. Kiến trúc chung của FPGA (Hình 2.2) dựa trên SRAM cơ bản có các thành phần cơ bản:

- Các khối lô-gic có thể cấu hình được
- Các khối vào ra có thể cấu hình được
- Mạng liên kết có thể lập trình được
- Các khối RAM

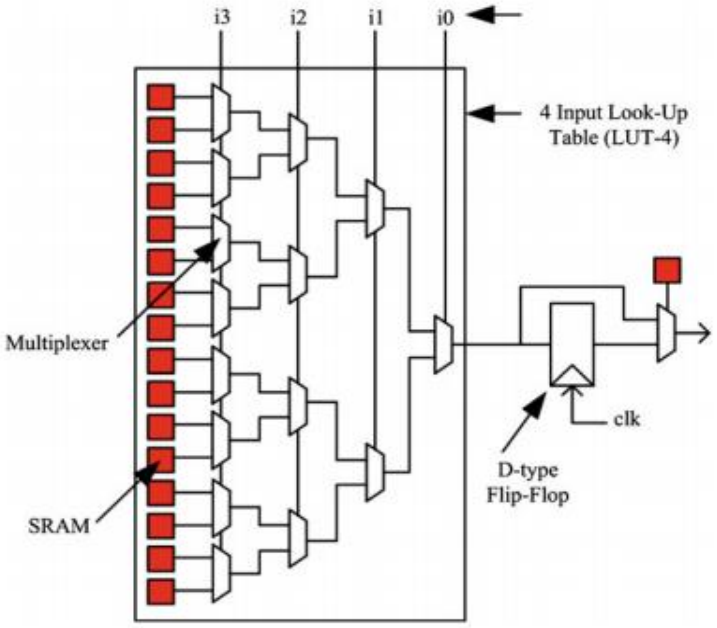


Hình 2.2: Tổng thể về kiến trúc FPGA [3].

Ngoài ra có thể còn có các mạch điều khiển các tín hiệu xung nhịp số phân phối cho từng khối lô-gic, khối vào ra và các khối mạch lô-gic bổ sung như các ALU, bộ nhớ, bộ ghép kênh (MUX), thanh ghi dịch, mạch giải mã.

Khối lô-gic có thể tái cấu hình (CLB): Mục đích của việc lập trình khối lô-gic trong FPGA là để cung cấp các tính toán và các phần tử nhớ cơ bản được sử dụng trong hệ thống số. Các khối lô-gic này là tài nguyên chính của một mạch FPGA. Trong hầu hết các mạch FPGA, mỗi một khối lô-gic chứa một số các mảnh, mà mỗi mảnh lại chứa các ô lô-gic với một số thành phần nhớ. Mỗi ô lô-gic được cấu hình để thực hiện các chức năng hàm lô-gic cơ bản như AND, OR, NOT được thực hiện trên các tín hiệu số nhờ sử dụng bảng ánh xạ

(LUT). Các khối lô-gic được liên kết với nhau qua mạng liên kết có thể lập trình được để tạo thành một mảng các khối lô-gic thực hiện các tác vụ tính toán lớn hơn.



Hình 2.3: Khối lô-gic cơ bản [8].

Bảng ánh xạ (LUT): Bảng ánh xạ giống như một RAM nhỏ, cũng được gọi là bộ tạo chức năng, được sử dụng để thực hiện các chức năng lô-gic nhờ cất giữ trạng thái lô-gic đầu ra đúng ở trong một vùng nhớ, mà trạng thái lô-gic ra tương ứng với từng tổ hợp của các biến vào. Bảng ánh xạ thường có bốn đầu vào có thể thực hiện bất kỳ chức năng lô-gic bốn đầu vào như hình 2.3.

Hệ thống mạch liên kết lập trình được: Các liên kết trong một mạch FPGA dùng để liên kết các khối lô-gic và các khối vào/ra lại với nhau để tạo thành một thiết kế. Các mạch liên kết bao gồm các bộ ghép kênh, transistor và bộ đệm ba trạng thái. Các transistor và bộ ghép kênh được dùng trong các khối lô-gic ùng để kết nối các khối lô-gic lại với nhau, ngoài ra hai thành phần trên cùng bộ đệm ba trạng thái còn được dùng cho các cấu trúc định tuyến bên trong mạch FPGA.

Khối ra vào: Khối vào ra cung cấp giao tiếp giữa các khối lô-gic và kiến trúc định tuyến đến các thành phần bên ngoài. Như vậy từng chân vào/ra của mạch FPGA có thể được lập trình để đảm bảo các giao tiếp điện cần thiết cho kết nối FPGA với hệ thống mà nó là thành phần. Cho nên một trong những vấn đề quan trọng nhất trong thiết kế kiến trúc

vào/ra là việc lựa chọn các tiêu chuẩn điện áp cung cấp và điện áp tham chiếu sẽ được hỗ trợ.

Khối RAM: là khối có dung lượng vài Kb, được nhúng ở các vị trí cố định trên FPGA để lưu trữ dữ liệu (không được sử dụng để thực hiện các chức năng lô-gic khác).

Các FPGA thường được lập trình sau khi đã hàn gắn trên bảng mạch in, tương tự như các cấu kiện lô-gic khả trình phức tạp lớn. Trên thực tế, bộ nhớ truy cập ngẫu nhiên tĩnh (SRAM) của mạch FPGA được lập trình lại mỗi khi bật nguồn, vì FPGA là dạng chip nhớ tạm thời. Do đó, cấu hình sẽ phải nạp lại mỗi lần khởi động lại chip FPGA. Muốn lưu giữ lại cấu hình đã lập trình cho FPGA thì ta phải mắc thêm một bộ nhớ chỉ đọc khả trình (PROM) hay bộ nhớ chỉ đọc lập trình được và xóa được (EEPROM) ngoài để có thể nạp cấu hình tự động trong quá trình hoạt động. Bộ nhớ ngoài này có nhiệm vụ lưu tệp cấu hình ở dạng nhị phân và mỗi khi thiết bị được bật nguồn tệp cấu hình từ EEPROM sẽ được nạp tự động vào bộ nhớ SRAM của mạch FPGA, FPGA hoạt động theo cấu hình đã được nạp đó. Các giao tiếp cấu hình cơ bản được nạp qua giao diện cổng JTAG và có thể được nạp lại cũng như cấu hình lại vô số lần thông qua giao diện đó.

2.4 Ứng dụng của mảng cổng lô-gic lập trình được

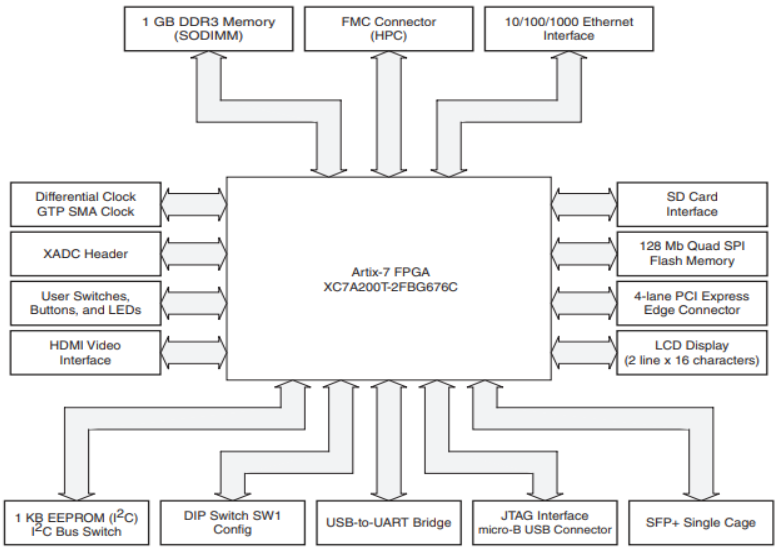
Ứng dụng của mạch FPGA bao gồm: xử lý tín hiệu số, các hệ thống hàng không vũ trụ, quốc phòng, tiền thiết kế mẫu ASIC, các hệ thống điều khiển trực quan, phân tích nhận dạng ảnh, nhận dạng giọng nói, mật mã học, mô hình phần cứng máy tính. Ngày nay FPGA còn được ứng dụng trong các kiến trúc xử lý song song, đặc biệt là các ứng dụng về mã hóa và giải mã.

Do có hiệu năng cao cùng khả năng tái cấu hình cho nên công nghệ FPGA có thể giải quyết được các công thức tính toán phức tạp mà trước kia chỉ được thực hiện qua các phần mềm trên máy tính. Cùng với đó công nghệ FPGA còn được ứng dụng cả trong các hệ thống thời gian thực. Tiếp theo học viên sẽ giới thiệu về Kit Artix-7 được sử dụng trong đề tài này.

2.5 Giới thiệu về Kit Artix 7 – AC701

Kít phát triển Xilinx Artix-7 FPGA có hiệu suất hệ thống cao cung cấp cho bạn khả năng tiền thiết kế (thiết kế mẫu) nhanh cho các ứng dụng đắt tiền. Bao gồm các thành phần cơ bản về phần cứng, công cụ thiết kế, IP, và các thiết kế tham khảo. Kit cũng bao gồm tính năng thiết kế tham khảo hướng mục tiêu cho phép khả năng kết nối nối tiếp hiệu suất

cao và giao tiếp bộ nhớ tiên tiến. Kiến trúc tổng thể của Artix-7 AC701 sẽ được biểu diễn dưới hình:



Hình 2.4: Sơ đồ khối Artix-7 [1].

Các thông số chủ yếu của kit Artix-7 bao gồm:

FPGA: Artix-7 XC7A200T-2FBG676C FPGA

Kit AC701 tương thích ROHS bao gồm FPGA XC7A200T-2FBG676C

Cấu hình:

Mạch cấu hình trên bo cho phép cấu hình qua cổng USB

JTAG header cung cấp sử dụng với cáp download: Platform Cable USB II

Quad SPI Flash: 32MB (256Mb)

Bộ nhớ:

DDR3 SODIMM 1GB up to 533MHz / 1066Mbps

Quad SPI Flash: 32MB (256Mb)

IIC EEPROM: 8Kb

Khe thẻ SD

Giao tiếp và mạng:

10/100/1000 Mbps Ethernet

SFP cage

Cổng GTP (TX, RX) với 4 đầu nối SMA

Cầu UART tới USB

Đầu nối cạnh PCI Express 4-lane

Hiển thị:

Đầu ra HDMI video

Hiển thị LCD (2x16)

LED (x4)

Đầu nối mở rộng:

Đầu nối FMC-HPC

2 bộ truyền nhận GTP

Cấp nguồn 1.8V, 2.5V, or 3.3V

PMOD (1x6 0.1" Header)

Xung nhịp:

Bộ dao động 200Mhz

Bộ dao động (Dải: 10MHz - 810 MHz, 156.250 MHz) cho I2C lập trình được

Các đầu vào khác để truy nhập

Điều khiển & I/O:

5 nút nhấn

Công tắc DIP

I/O người dùng SMA

AMS FAN Header (2 I/O)

7 chân vào/ra qua đầu đọc LCD.

Nguồn

AC Power adapter (12V)

Tương tự

XADC header

Ngoài ra còn có bảng so sánh Artix-7 với các dòng khác của Xilinx được thể hiện trong bảng 2.1 [4]:

Bảng 2.1: Bảng so sánh một số dòng bo mạch FPGA

Max. Capability	Spartan-7	Aritix-7	Kintex-7	Virtex-7
Cổng lô-gic	102K	215K	478K	1.955K
Dung lượng RAM	4,2Mb	12Mb	34Mb	68Mb
DSP Slices	160	740	1.920	3.600
Hiệu năng DSP	176 GMAC/s	929 GMAC/s	2.845 GMAC/s	5.335 GMAC/s
MicroBlaze CPU	260 DMIPs	303 DMIPs	438 DMIPs	441 DMIPs
Cổng thu/phát	-	16	32	96
Tốc độ thu/phát	-	6,6 Gb/s	12,5 Gb/s	28,5 Gb/s
Băng thông nối tiếp	-	211 Gb/s	800 Gb/s	2.784 Gb/s
Giao diện PCle	-	x4 Gen2	x8 Gen2	x8 Gen3
Giao diện bộ nhớ	800 Mb/s	1.066 Mb/s	1.866 Mb/s	1.866 Mb/s
Cổng I/O	400	500	1200	1200
Điện năng cung cấp I/O	1,2V-3,3V	1,2V-3,3V	1,2V-3,3V	1,2V-3,3V

Qua các thông số trên có thể thấy được Kit Artix-7 AC701 đáp ứng đủ nhu cầu sử dụng về cả bộ nhớ tốc độ xử lý lẫn hiệu năng để tạo ra một hệ thống SoC trước khi đưa vào sản xuất chính. Ngoài ra, vi xử lý được sử dụng trong đề tài này là Microblaze của hãng Xilinx. Tiếp theo chúng ta sẽ tìm hiểu về cấu trúc cũng như là hiệu năng của vi xử lý này.

2.6 Vi xử lý Microblaze

Các bộ vi xử lý có sẵn dùng cho dòng FPGA (Field Programmable Gate Arrays) của Xillinx sử dụng với các công cụ phần mềm có trong phần mềm EDK (Embedded Development kit) được phân thành hai loại: bộ vi xử lý mềm Microblaze và bộ vi xử lý cứng đã được nhúng sẵn Power PC. Microblaze là bộ vi xử lý được dùng hầu hết trong FPGA các dòng như Spartan-II, Spartan-III, Virtex, Artix của hãng Xillinx. Microblaze là một vi điều khiển ảo, nó tồn tại dưới dạng phần mềm đã được phát triển của hãng Xillinx, nhà thiết kế có thể thiết lập các thông số để sử dụng đối với vi điều khiển này (UART, các cổng vào ra ngoại vi, ..) thông qua phần mềm EDK. MicroBlaze là bộ xử lý mềm nhúng 32-bit của Xillinx. Tập lệnh rút gọn RISC, với các bus riêng biệt để truy xuất dữ liệu và lệnh từ bộ nhớ trên chip và bộ nhớ ngoài tại cùng một thời điểm.

2.6.1 Kiến trúc vi xử lý Microblaze

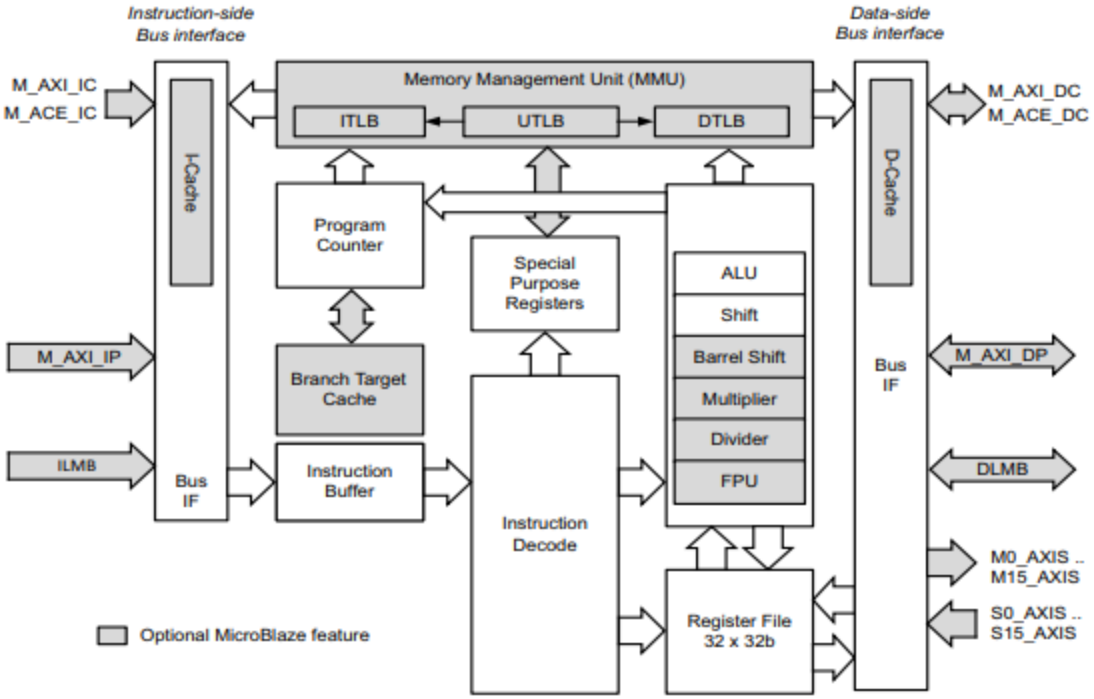
Kiến trúc của MicroBlaze có các đặc điểm sau: tập lệnh 32 bit với 3 toán hạng và 2 chế độ định địa chỉ. Đường bus 32 bit địa chỉ, một khối ghi dịch, hai cấp độ ngắt, khối tính toán số học ALU: gồm các bộ cộng/trừ, ghi dịch/lô-gic, nhân. Trong đó có các thành phần sau:

Đường bus giao tiếp
Bộ đệm lệnh
Bộ giải mã lệnh
Bộ đếm chương trình
Khối cộng/trừ
Khối dịch/lô-gic
Bộ nhân
Tập thanh ghi dữ liệu gồm 32 thanh ghi 32-bit

MicroBlaze cung cấp 3 giao tiếp bộ nhớ:

Bus bộ nhớ cục bộ
Bus xử lý cục bộ
Bus kết nối ngoại vi

Mối quan hệ giữa các thành phần trong vi xử lý được thể hiện trong hình 2.5:



Hình 2.5: Sơ đồ khối kiến trúc vi xử lý Microblaze 32 bits [2].

Các giao tiếp với bộ nhớ đệm cũng được chia ra bus kết nối với lệnh bộ nhớ đệm và kết nối dữ liệu với bộ nhớ đệm. Bus vào/ra đầu tiên của Microblaze là giao diện kết nối AXI, đây là bus kết nối giữa hệ thống và bộ nhớ với quan hệ chủ/tớ. Các giao tiếp đồng thời xử lý được hỗ trợ bởi các kết nối AXI4-Stream. Các kết nối ngoại vi được thực hiện qua các bus ngoại vi trên chip theo chuẩn IBM, chuẩn này cung cấp ba loại bus giành cho kết nối liên tục đa lõi, các thư viện macro và cổng lô-gic cho người dùng:

Bus xử lý cục bộ
Bus ngoại vi trên chip
Bus điều khiển thanh ghi

Với bus ngoại vi trên chip có chia ra: Giao diện lệnh và giao diện dữ liệu. Toàn bộ các kết ngối ngoại vị qua bus OPB bao gồm:

Đồng hồ
Hẹn giờ/bộ đếm
Điều khiển Interrupt
Điều khiển SRAM
Điều khiển bộ nhớ flash

Điều khiển bộ nhớ ZBT
Điều khiển BRAM
Điều khiển bộ nhớ DDR
Điều khiển SDRAM
Giao diện kết nối UART
Cổng vào/ra
Giao diện kết nối SPI
Giao diện I2C
Giao diện UART 16450/550
Giao diện kết nối mạng Ethernet 10/100 MAC

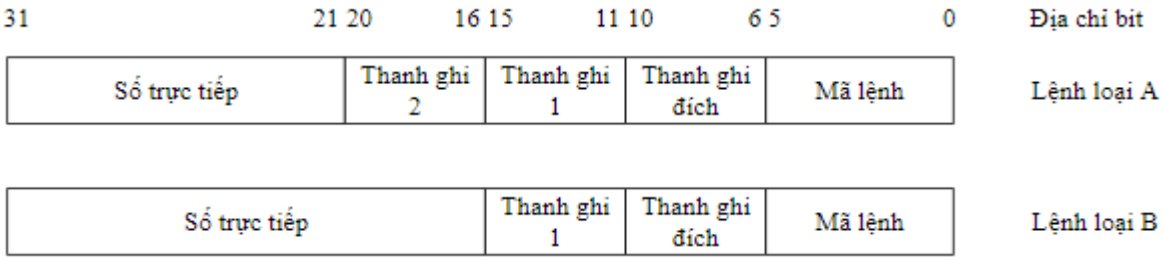
2.6.2 Định dạng dữ liệu và kiến trúc tập lệnh của Microblaze

Tập lệnh rút gọn RISC của Microblaze sử dụng hai định dạng lệnh 32-bit:

Định dạng loại A: là tập các lệnh giữa thanh ghi – thanh ghi. Nó gồm có trường opcode, một thanh ghi toán hạng đích và hai thanh ghi nguồn.

Định dạng loại B: được sử dụng các lệnh thanh ghi – số. Nó gồm có trường opcode, một thanh ghi đích và một thanh ghi nguồn, và một giá trị nguồn số trực tiếp 16-bit.

Định dạng tập lệnh của Microblaze sẽ được biểu diễn cụ thể như sau:



Hình 2.6: Định dạng lệnh trong Microblaze.

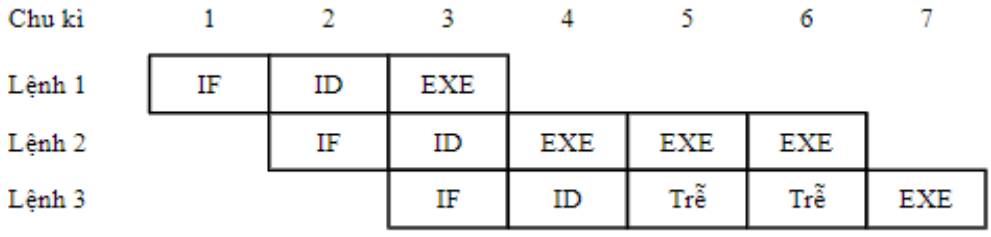
Microblaze sử dụng kiến trúc Load/Store của RISC trong đó có thể truy cập đến bộ nhớ theo ba kích thước dữ liệu: 8 bits, 16 bits, và 32 bits. Thành phần chính của kiến trúc RISC trên vi xử lý Microblaze dựa vào tập thanh ghi 32-bit bảng ánh xạ bộ nhớ truy cập ngẫu nhiên với các tập lệnh truy cập riêng bộ nhớ dữ liệu và bộ nhớ lệnh. Microblaze hỗ trợ kết nối các khối bộ nhớ truy cập ngẫu nhiên (BRAM) cả trong chip và ngoài chip.

Cũng như các vi xử lý kiến trúc RISC khác Microblaze sử dụng kiến trúc đường ống. Đối với hầu hết các câu lệnh, mỗi giai đoạn mất một chu kì để hoàn thành. Vì vậy số chu

kỳ để hoàn thành một câu lệnh cụ thể bằng với số giai đoạn trong cấu trúc đường ống và một câu lệnh được hoàn thành trên mỗi chu kì trong trường hợp không có dữ liệu, điều khiển và lỗi về cấu trúc. Một nguy cơ về xung đột dữ liệu xảy ra khi kết quả của một lệnh là cần thiết với lệnh tiếp đó. Điều này dẫn đến việc đình trệ trong quá trình xử lý trừ khi kết quả có thể được chuyển tiếp cho câu lệnh tiếp theo. Trình biên dịch GNU của Microblaze sẽ sắp xếp các câu lệnh một các tối ưu hóa để tránh sự xung đột này. Một xung đột điều khiển xảy ra khi một câu lệnh đã được thực hiện nhưng câu lệnh tiếp theo không có sẵn để thực hiện ngay lập tức. Microblaze cung cấp các khe trễ và bộ đệm để giảm số chu kì trống. Một nguy cơ cấu trúc xảy ra đối với một vài lệnh yêu cầu nhiều chu kỳ trong giai đoạn thực hiện hoặc giai đoạn sau để hoàn thành. Điều này đạt được bằng cách trì hoãn các đường ống. Tải và lưu trữ các lệnh truy cập bộ nhớ chậm hơn có thể mất nhiều chu kỳ. Các giai đoạn bị đình trệ cho đến khi truy cập hoàn thành. MicroBlaze cung cấp bộ đệm tiền nhận lệnh để làm giảm xung đột tiềm ẩn trong bộ nhớ khi thực hiện các lệnh đa chu kỳ. Trong khi đường ống bị đình trệ vì bất kỳ lý do nào khác, bộ đệm tiền nhận lệnh tiếp tục tải các lệnh tuần tự ngay sau đó. Và khi kiên trúc đường ống thực thi trở lại thì giai đoạn nhận lệnh có thể lấy lệnh trực tiếp từ bộ đệm tiền nhận lệnh, thay vì đợi cho bộ nhớ lệnh truy cập để thi hành. Nếu các câu lệnh bị sửa đổi trong quá trình thực thi thì bộ đệm tiền nhận lệnh cũng nên được làm trống để đảm bảo nó không chưa các câu lệnh cũ.

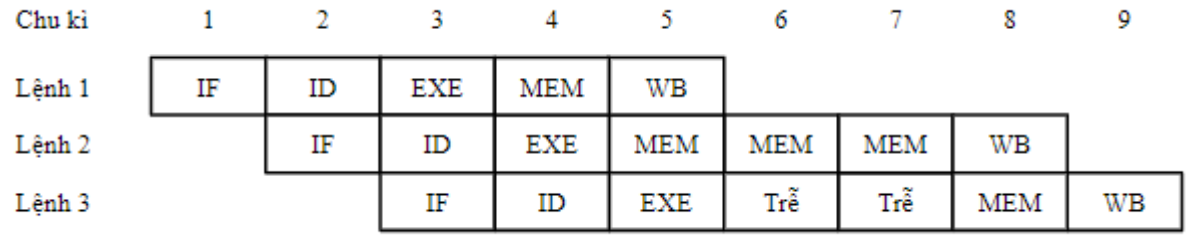
Vi xử lý Microblaze tích hợp được ba loại cấu trúc đường ống cho ba mục đích khác nhau.

Với kiến trúc tiết kiệm diện tích thì kiến trúc đường ống được thiết kế với ba giai đoạn để tiết kiệm chi phí phần cứng: đọc lệnh (IF), giải mã lệnh (ID), và thực thi lệnh (EXE). Các quá trình đẩy dữ liệu, trễ trong đường ống và rẽ nhánh được thực hiện tự động trong phần cứng. Hình 2.7 thể hiện kiến trúc đường ống ba giai đoạn của Microblaze:



Hình 2.7: Kiến trúc đường ống ba giai đoạn.

Kiến trúc đường ông còn được thiết kế với năm giai đoạn để tối đa hóa hiệu năng thực thi: Ngoài ba giai đoạn ở trên kiến trúc này còn có thêm giai đoạn: truy cập bộ nhớ (MEM), ghi kết quả cuối vào thanh ghi (WB) được thể hiện ở trong hình 2.8:

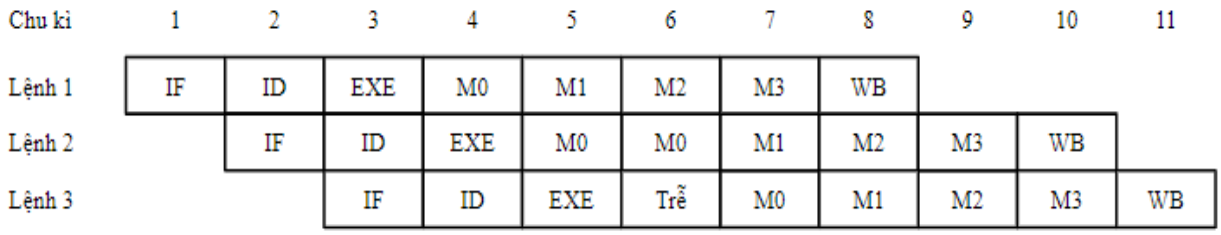


Hình 2.8: Kiến trúc đường ống năm giai đoạn.

Kiến trúc này có hai loại xung đột dữ liệu [2]:

- Một lệnh ở giai đoạn giải mã (ID) đang cần kết quả từ một lệnh đang trong quá trình thực thi (EXE) dưới dạng toán hạng nguồn. Trong trường hợp này các loại lệnh ở giai đoạn thực thi bao gồm tải, lưu, dịch, nhân, chia và lệnh dấu phẩy động. Kết quả dẫn đến việc 1-2 chu kì trễ.
- Một lệnh ở giai đoạn giải mã (ID) cần sử dụng kết quả một lệnh đang ở giai đoạn truy cập bộ nhớ (MEM) làm toán hạng nguồn. Trong trường hợp này các lệnh ở giai đoạn truy cập bộ nhớ là tải, nhân và dấu phẩy động. Kết quả cần 1 chu kỳ trễ.

Ngoài ra còn có chế độ thứ ba chính là tối đa hóa tần số hoạt động có thể chính là kiến trúc đường ống tám giai đoạn bao gồm: đọc lệnh (IF), giải mã lệnh (ID), thực thi lệnh (EXE), truy cập bộ nhớ 0 (M0), truy cập bộ nhớ 1 (M1), truy cập bộ nhớ 2 (M2), truy cập bộ nhớ 3 (M3) và ghi kết quả cuối vào thanh ghi (WB). Kiến trúc này được mô tả dưới hình 2.9:



Hình 2.9: Kiến trúc đường ống tám giai đoạn.

Với kiến trúc này thì có bốn loại xung đột dữ liệu [2]:

- Khi một câu lệnh đang ở giai đoạn giải mã (ID) cần kết quả từ câu lệnh thực thi (EXE) như một toán hạng nguồn. Trong trường hợp này các loại lệnh thực thi bao gồm tải, lưu, dịch, nhân, chia và lệnh dấu phẩy động. Kết quả dẫn đến việc 1-5 chu kì trễ.
- Một lệnh đang ở giai đoạn giải mã (ID) cần sử dụng kết quả một lệnh ở bộ nhớ M0 làm toán hạng nguồn. Trong trường hợp này các lệnh ở bộ nhớ M0 là tải, nhân và dấu phẩy động. Kết quả cần 1-4 chu kỳ trễ.
- Một lệnh đang ở giai đoạn giải mã (ID) cần sử dụng kết quả một lệnh ở bộ nhớ M0 và M1 làm toán hạng nguồn. Trong trường hợp này các lệnh ở bộ nhớ M0 và M1 là tải, nhân và dấu phẩy động. Kết quả cần 1-3 hoặc 1-2 chu kỳ trễ.
- Khi một câu lệnh đang ở giai đoạn giải mã (ID) cần kết quả từ câu lệnh ở bộ nhớ M3 như một toán hạng nguồn. Trong trường hợp này các loại lệnh thực thi bao gồm lệnh tải và lệnh dấu phẩy động. Kết quả dẫn đến việc 1 chu kì trễ.

Sau khi đã tìm hiểu về cấu trúc mảng cổng lô-gic có khả năng lập trình cũng như các thành phần cơ bản của nó để có thể thiết kế nên một hệ thống hoàn chỉnh dựa trên công nghệ này. Tiếp theo học viêc sẽ tiến hành phần tích cũng như thiết kế nên một hệ thống xử lý ảnh được sử dụng công cụ thiết kế, mô phỏng FPGA đã phân tích ở trên.

Chương 3

ĐỀ XUẤT VÀ THỰC THI THỬ NGHIỆM HỆ THỐNG XỬ LÝ ẢNH TRÊN ARTIX-7

Với sự phát triển ngày càng mạnh trong ngành công nghệ xử lý ảnh để đưa vào ứng dụng vào rất nhiều ứng dụng khác nhau hiện nay cùng với đó là sự đi liền với tính linh hoạt và hiệu năng hoạt động của một hệ thống phần cứng để vừa đảm bảo được hiệu năng hoạt động lại vừa có thể chỉnh sửa trực tiếp thuật toán trên hệ thống. Thì một hệ thống FPGA có thể đáp ứng được cả hai vấn đề đó là có thể chỉnh sửa trực tiếp trên hệ thống cũng như có thể cải thiện hiệu năng hoạt động của nó. Tuy nhiên để thực hiện chỉnh sửa trên phần cứng thì cũng cần sự trợ giúp của các phần mềm trung gian như các phần mềm của nhà sản xuất Xilinx, Althena hoặc bên thứ ba như Synopsys. Trong đó Kit Artix-7 của Xilinx giúp cho vấn đề thiết kế hệ thống được đơn giản hơn khi có công cụ hỗ trợ cùng thông số hợp lý và phù hợp với giá thành để tạo nên một hệ thống mẫu trước khi đưa vào sản xuất.

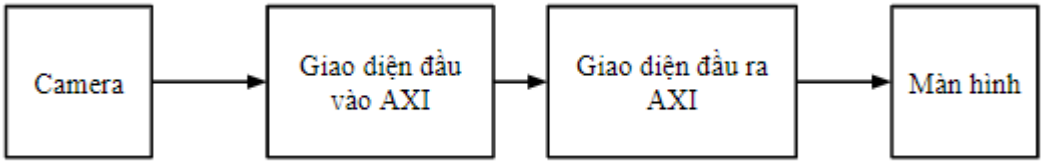
3.1 Công cụ Vivado Design Suite

Vivado Design Suite là tổ hợp các phần mềm của hãng Xilinx. Phần mềm này được tạo ra bằng việc nâng cấp các thế hệ phần mềm thiết kế cũ ISE Design Suite. Vivado được dùng để phát triển các ứng dụng trên thế hệ chip và board Xilinx® đời 7, Zynq®-7000…. Các thế hệ board cũ của Xilinx như Series 6 sẽ vẫn được hỗ trợ bởi ISE, Plan Ahead ... Vivado là phần mềm có rất nhiều chức năng. Nói một cách ngắn gọn, nó hỗ trợ tất cả các khâu của quá trình thiết kế lô-gic sử dụng FPGA.

3.2 Sơ đồ khối liên kết camera với Artix 7

Để thực hiện liên kết với Artix-7 thì camera sử dụng ở đây là VITA-2000. Bước đầu tiên trong một hệ thống xử lý ảnh chính là phải kết nối camera với hệ thống FPGA và đưa ảnh ra màn hình hiển thị. Ảnh được thu nhận ở đây chính là ảnh với chất lượng 1080p. Để

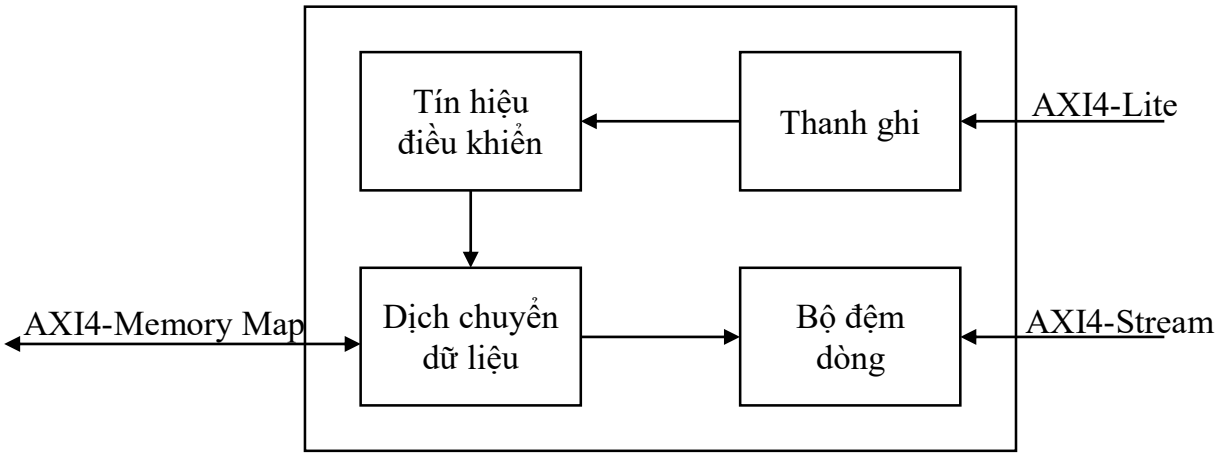
thực hiện ghép nối với hệ thống FPGA thì ta thực hiện kết nối camera với bus của Kit Artix-7 mà cụ thể ở đây chính là AXI bus qua sơ đồ dưới đây:



Hình 3.1: Kết nối camera-vita 2000 với màn hình.

Đây là bước tiền đề để tạo nên một hệ thống xử lý ảnh hoàn chính. Bởi vậy ngoài việc kết nối và truyền được dữ liệu camera lên truyền hình ta còn phải đảm bảo dữ liệu truyền đi không bị mất cũng như bị sai lêch và đỗ trễ cũng phải rất ít vì hệ thống làm việc với thời gian thực. Khi tín hiệu xuất ra màn hình được chính xác và sắc nét thì chúng ta có thể đảm bảo sự liên kết với bus của Kit FPGA đã hoàn thành và sau đó sẽ là bước tiếp theo đưa dữ liệu RAW từ camera vào trong các IP để tiến hành xử lý ảnh. Ở bước này dữ kiệu sẽ được đi qua khối truy cập trực tiếp bộ nhớ video (VDMA) sau đó mới được xuất ra màn hình.

Khối VDMA thực hiện tối ưu hóa video hiệu năng cao với bộ đệm khung và đặc tính truy cập bộ nhớ trực tiếp. Cùng với đó kết nối AXI và AXI MIG triển khai bộ điều khiển bộ nhớ đa cổng để chia sẻ dữ liệu từ nhiều nguồn thông qua một thiết bị bộ nhớ chung, điển hình là DDR, SDRAM. Khối VDMA chuyển các luồng dữ liệu video được đệm vào từ bộ nhớ và hoạt động dưới chế độ điều khiển phần mềm động hoặc chế độ cấu hình tĩnh. Sơ đồ khối kiến trúc khối VDMA được thể hiện trong Hình 3.2:



Hình 3.2: Kiến trúc khối truy cập trực tiếp bộ nhớ video [2].

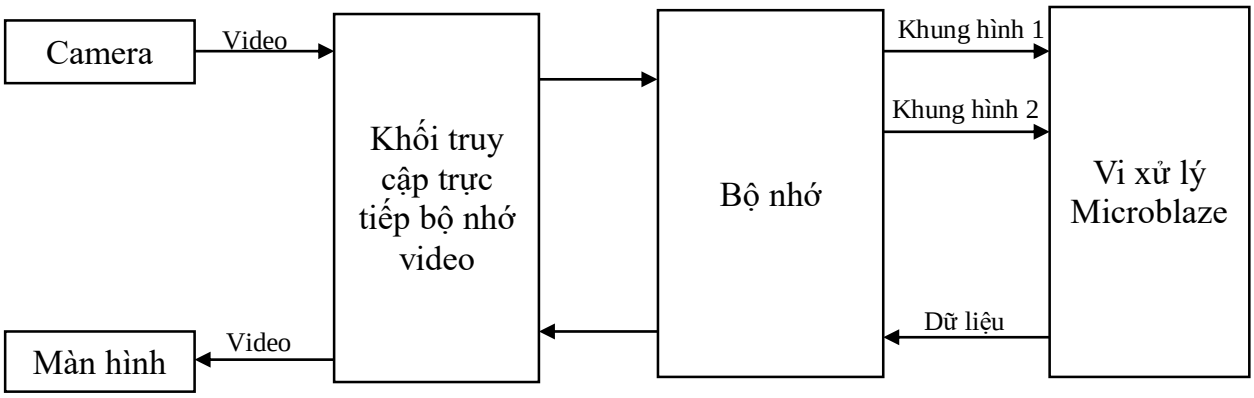
Sau khi các thanh ghi được lập trình thông qua giao diện AXI4-Lite, khối điều khiển và trạng thái tạo các lệnh thích hợp cho khối dịch chuyển dữ liệu khởi tạo các lệnh "Viết" và "Đọc" trên giao diện AXI4 Master. Bộ đệm dòng không đồng bộ có thể cấu hình để tạm thời giữ dữ liệu điểm ảnh trước khi ghi ra giao diện AXI4-Memory Map hoặc giao diện AXI4-Stream. Trong lệnh "Viết", khối truy cập trực tiếp bộ nhớ video chấp nhận các khung ảnh trên giao diện AXI4-Stream Slave và ghi nó vào bộ nhớ hệ thống bằng giao diện AXI4 Master. Trong lệnh "Đọc", khối truy cập trực tiếp bộ nhớ video sử dụng giao diện AXI4 Master để đọc các khung hình từ bộ nhớ hệ thống và xuất nó trên giao diện AXI4-Stream Master. Cả hai lệnh "Viết" và "Đọc" hoạt động độc lập với nhau. Khối truy cập trực tiếp bộ nhớ video cũng cung cấp các tùy chọn để đồng bộ hóa các khung ảnh vào/ra với tín hiệu đồng bộ hóa bên ngoài.

Ngoài ra còn có một số đánh giá về hiệu năng của khối VDMA trên một số bo mạch FPGA đượng liệt kê trong bảng 3.1 [7]:

Bảng 3.1: Hiệu năng của AXI VDMA trên một số bo mạch FPGA

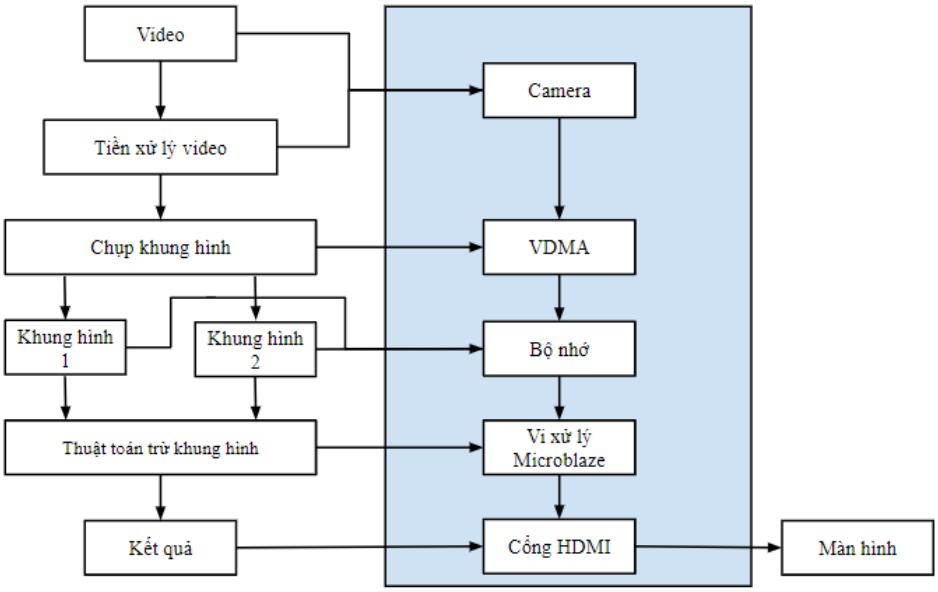
Bo mạch	Lớp tốc độ	Fmax (MHz)		
		AXI4	AXI-Stream	AXI-Lite
Virtex®-7	-1	200	200	180
Kintex®-7		200	200	180
Artix®-7		150	150	120
Virtex®-7	-2	240	240	200
Kintex®-7		240	240	200
Artix®-7		180	180	140
Virtex®-7	-3	280	280	220
Kintex®-7		280	280	220
Artix®-7		200	200	160

Sau khi đã thực hiện được lưu trữ và xử lý dữ liệu đầu vào chúng ta sẽ đưa tín hiệu vào trong vi xử lý để tiến hành bước quan trọng nhất của hệ thống đó là tiến hành thuật toán vào dữ liệu để thực hiện các ứng dụng của xử lý ảnh như phát hiện chuyển động. Khi đó dữ liệu đưa vào bộ nhớ sẽ phân theo từng luồng khung hình đưa vào vi xử lý để tiến hành xử lý hình ảnh và được đưa lại về khối VDMA để xuất ra màn hình. Các bước thực hiện sẽ được mô tả như hình 3.3:



Hình 3.3: Xử lý hình ảnh trên Microblaze.

Khi đó hình ảnh từ camera sẽ được đưa qua khối truy cập trực tiếp bộ nhớ video và đưa vào bộ nhớ của hệ thống. Dữ liệu khi đi qua khối VDMA sẽ được xử lý và chụp lại tạo thành các khung hình và sau đó được lưu vào trong bộ nhớ. Từ đây bộ nhớ sẽ chia thành các khung hình để đưa vào khối xử lý Microblaze. Microblaze sẽ đưa ra các tín hiệu điều khiển về nhận và xử lý dữ liệu để đưa hình ảnh đến một IP để tiến hành xử lý hình ảnh theo mong muốn. Từ đó chúng ta sẽ có được kết quả dữ liệu truyền tín hiệu về cho Microblaze. Sau đó microblaze sẽ đưa một tín hiệu để đưa dữ liệu vào bộ nhớ và đưa về khối VDMA. VDMA sẽ có chức năng xử lý hình ảnh đầu ra và đưa về màn hình chính là kết quả đã xử lý hình ảnh của camera đầu vào. Các bước xử lý ảnh xử sẽ được thực hiện theo sơ đồ khối dưới đây:

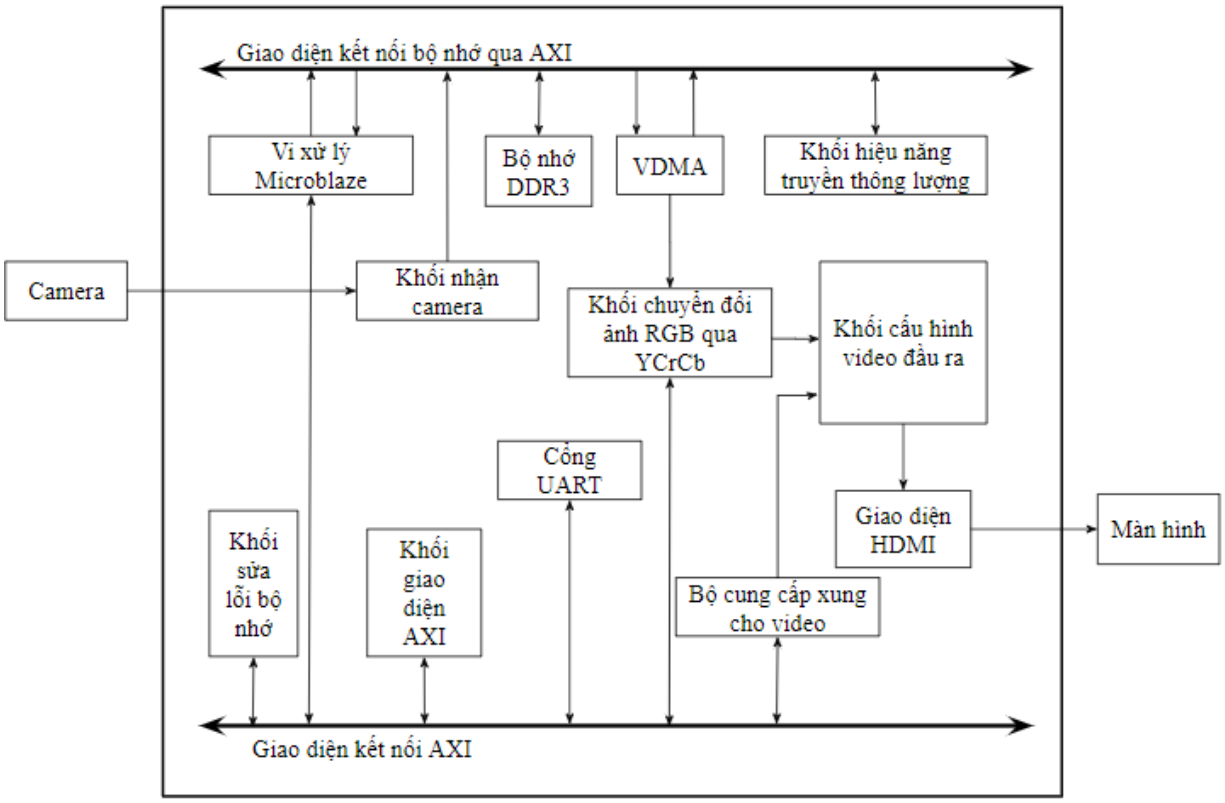


Hình 3.4: Mối liên hệ giữa kiến trúc phần mềm và phần cứng trong hệ thống.

Sau khi đã thiết kế được sơ đồ khối các kết nối cơ bản thì chúng ta thực hiện tạo hệ thống hoàn chỉnh trên Kit Artix-7.

3.3 Hệ thống xử lý ảnh đề xuất trên FPGA sử dụng Kit Artix-7

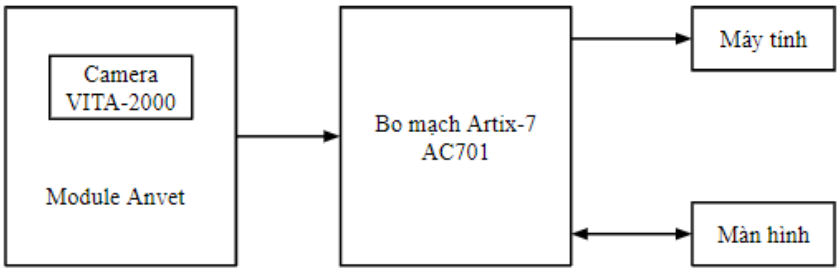
Với việc thực thi trên công cụ Vivado và các khối có sẵn để thiết lập nên một hệ thống xử lý ảnh hoàn chỉnh chạy trên thời gian thực thì học viên đề xuất một hệ thống xử lý ảnh như hình:



Hình 3.5: Hệ thống xử lý ảnh để xuất trên Artix-7 AC701.

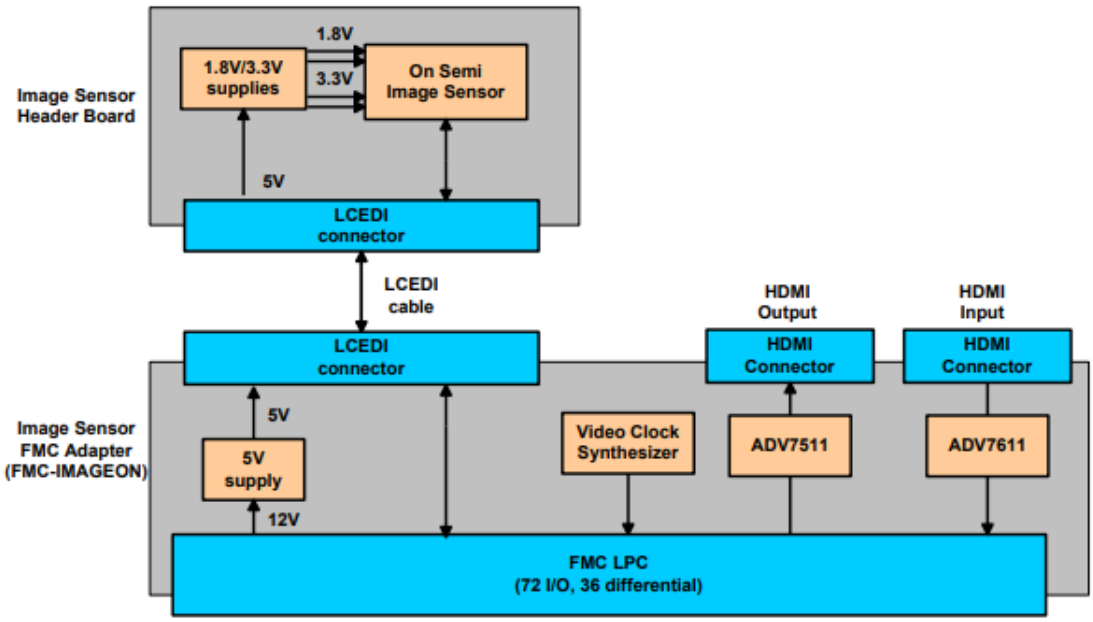
Trong thiết kế này chứa hai giao diện kết nối AXI, mỗi liên kết này nhằm cân bằng thông lượng, diện tích và thời gian. Giao diện kết nối bộ nhớ dùng cho các chế độ chủ/tớ với tốc độ cao bao gồm thông lượng và tối ưu hóa tần số hoạt động tối đa. Giao diện kết nối AXI thường được dùng để tối ưu hóa cho các khu vực và sử dụng cho bộ vi xử lý để truy cập các thanh ghi tớ của các IP khác nhau. Ngoài ra nó có các tham số cho phép các bộ đệm vào trước ra trước (FIFO) cải thiện thông lượng và các thanh ghi cải thiện thời gian.

Đầu vào video được truyền bởi camera VITA-2000, được cấu hình cho độ phân giải 1080p. Video đầu vào được lấy từ camera VITA-2000 từ trong bộ chuyển đổi Anvet FMC-IMAGEON. Sơ đồ kết nối được thể hiện bằng sơ đồ khối như hình 3.5:



Hình 3.6: Sơ đồ kết nối Module Anvet và bo mạch Artix-7.

Kiến trúc cụ thể của bộ chuyển đổi Anvet FMC-IMAGEON được biểu diễn như Hình 3.7:

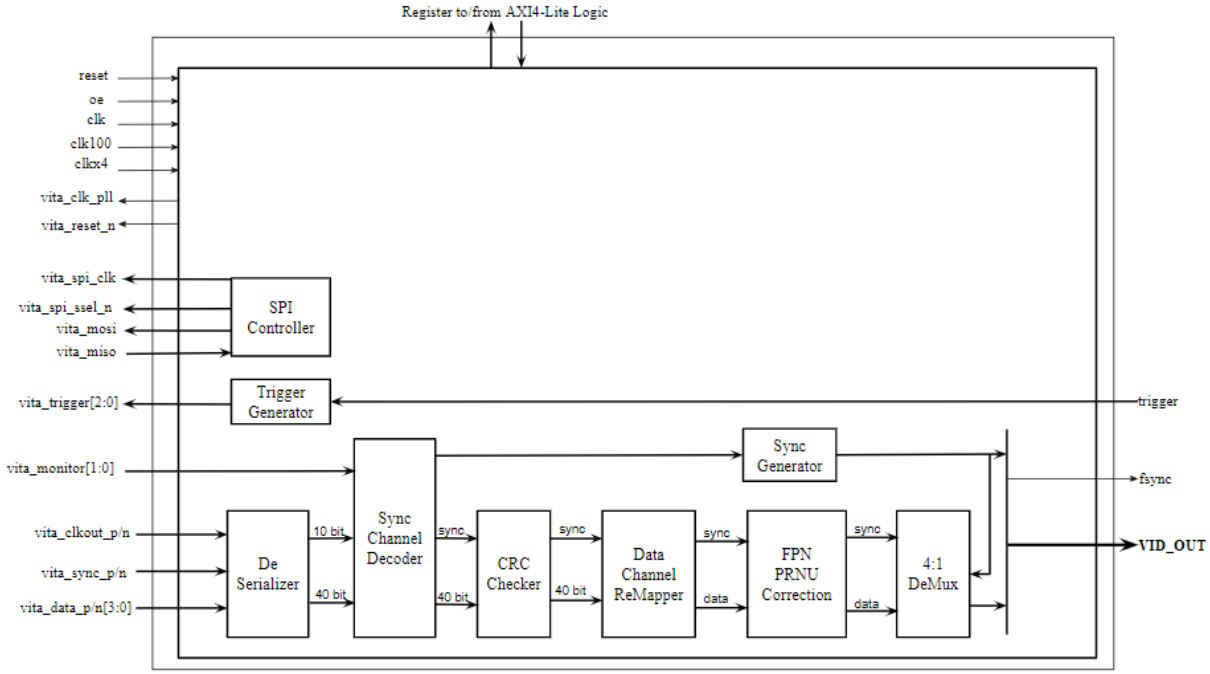


Hình 3.7: Sơ đồ khối kiến trúc bộ chuyển đổi Anvet [6].

Bộ chuyển đổi Anvet được kết nối với kit Artix-7 bằng chân FMC và bộ chuyển đổi sẽ lấy nguồn từ kit Artix-7 để nuôi toàn bộ thiết bị trong module. Camera VITA sẽ được kết nối với chân FMC để đưa dữ liệu vài kit Artix-7 để xử lý. Nó sẽ kết nối với chân FMC gián tiếp bằng kết nối giao diện hiển thì qua màn hình LCD (LCEDI). Ngoài dữ liệu video

thì bộ chuyển đổi Anvet còn đưa xung nhịp của video vào để thực hiện đồng bộ hóa với khối thu nhận video trong kit Artix-7.

Hình ảnh đầu vào là ảnh thô được chuyển đổi thành hình ảnh RGB bằng một đường ống xử lý hình ảnh được triển khai bằng lõi video để loại bỏ các điểm ảnh bị lỗi, khử màu và chỉnh màu cho hình ảnh. Bộ đệm khung video được triển khai trong bộ nhớ DDR3 của hệ thống xử lý, giúp hình ảnh có thể truy cập được vào lõi bộ xử lý Microblaze thông qua AXI VDMA. Bộ đệm khung video không cần thiết cho hoạt động của đường ống xử lý hình ảnh, nhưng được bao gồm trong thiết kế để cho phép chụp ảnh video đầu vào để phân tích. Hình ảnh sẽ đi qua khối camera đầu vào có kiến trúc như Hình 3.8:



Hình 3.8: Kiến trúc khối thu nhận camera VITA-2000

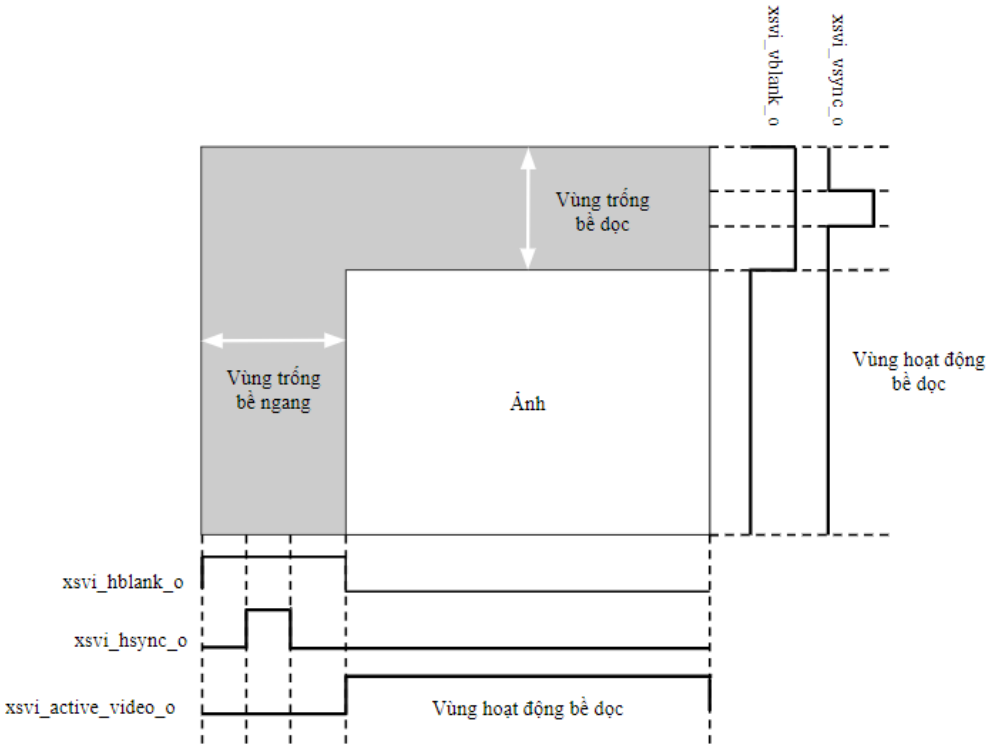
Khối thu nhận ảnh dùng thể thực hiện thu nhận hình ảnh từ camera VITA-2000. Khối De-Serializer dùng để thực hiện việc thu tín hiệu vi phân điện áp thấp. Khối "Sync Channel Decoder" dùng để giải mã kênh Sync và tạo các tín hiệu đồng bộ hóa phù hợp tín hiệu đồng bộ hóa sẽ được đưa qua khối "CRC Checker" để kiểm tra xem có lỗi đồng bộ hóa nào hay không. Khối "Sync Generator" tạo ra các tín hiệu đồng bộ hóa tiêu chuẩn VSYNC/HSYNC, VBLANK/HBlanK, DE cần thiết để tạo giao diện video theo thời gian chung (VID _OUT). Bộ phân kênh lấy 4 giá trị điểm ảnh song song đồng bộ với xung nhịp "clk" và tái tạo một luồng điểm ảnh đồng bộ hóa với xung nhịp "clkx4". Khối này có chứa FIFO có thể lưu trữ tối đa 6 dòng video đang hoạt động. Cổng fsync tạo ra xung nhịp hoạt

động trong một chu kỳ, khi bắt đầu mỗi khung hình mới. Nó có thể được sử dụng để đồng bộ hóa khi chuyển qua khối truy cập trực tiếp bộ nhớ video. Ngoài ra còn có chức năng của các tín hiệu vào ra trong khối thu nhận ảnh được thể hiện chi tiết trong bảng 3.2:

Bảng 3.2: Chức năng các tín hiệu trong khối thu nhận camera VITA-2000

Cổng	Hướng	Giao diện Bus	Chức năng
reset	In	Slave AXI	Reset
oe	In	Slave AXI	Cho phép đầu ra
clk	In	Slave AXI	Xung nhịp
Clk100	In	Slave AXI	Xung nhịp tần số 100 MHz
xsvi_active_video_o	Out	VID_OUT	Cho phép đưa dữ liệu ra
xsvi_vsync_o	Out	VID_OUT	Dữ liệu chiều dọc khung hình
xsvi_hsync_o	Out	VID_OUT	Dữ liệu chiều ngang khung hình
xsvi_vblank_o	Out	VID_OUT	Dữ liệu trống chiều dọc khung hình
xsvi_hblank_o	Out	VID_OUT	Dữ liệu trống chiều ngang khung hình
xsvi_video_data_o	Out	VID_OUT	Dữ liệu video đầu ra
vita_clk_pll	Out	VITA_CAMERA	Xung nhịp cung cấp cho camera VITA
vita_reset_n	Out	VITA_CAMERA	Tín hiệu reset mức thấp
vita_trigger[2:0]	Out	VITA_CAMERA	Tín hiệu trigger tới camera VITA
vita_monitor[1:0]	In	VITA_CAMERA	Lựa chọn màn hình từ camera VITA
vita_spi_sclk	Out	VITA_CAMERA	Xung nhịp SPI từ camera VITA
vita_spi_ssel_n	Out	VITA_CAMERA	Tín hiệu lựa chọn SPI tới camera VITA
vita_spi_mosi	Out	VITA_CAMERA	Tín hiệu SPI truyền vào camera VITA
vita_spi_miso	In	VITA_CAMERA	Tín hiệu SPI từ camera
vita_clk_out_p/n	In	VITA_CAMERA	Xung nhịp từ camera VITA
vita_sync_p/n	In	VITA_CAMERA	Tín hiệu đồng bộ hóa từ camera VITA
vita_data[3:0]_p/n	In	VITA_CAMERA	Dữ liệu từ camera VITA

Các tín hiệu đầu ra của khối thu nhận ảnh sẽ được biểu diễn như trong hình sau:



Hình 3.9: Sơ đồ thời gian các tín hiệu đầu ra trong khối thu nhận ảnh

Đây là khối quan trong nhất trong việc quyết đinh chất lượng hình ảnh được lưu trữ trong bộ nhớ. Khối này dẽ có chức năng chính trong việc đồng bộ hóa xung nhịp của camera VITA với bo mạch FPGA từ đó tạo khả năng liên kết trong việc thu nhận ảnh trong mỗi chu kì để không tạo ra xung đột trong việc khác xung nhịp trong các khối. Khối thu nhận này sẽ có các tín hiệu điều khiển cho phép nhận tín hiệu từ camera và sau khi thực hiện xong các thao tác xử lý về động bộ hóa, xử lí nhiễu cũng như liên kết các tín hiệu vi phân điện áp thấp của camera thì sẽ có tín hiệu cho phép truyền dữ liệu đi qua các khối khác để xử lý các giai đoạn tiếp theo.

Khối truy cập trực tiếp bộ nhớ video được thiết kế để cung cấp khả năng đọc/ghi video từ miền ánh xạ bộ nhớ AXI4 sang miền AXI4-Stream và ngược lại. Khối này cung cấp truyền dữ liệu tốc độ cao giữa bộ nhớ hệ thống và IP video đích dựa trên luồng AXI4. Lõi khối truy cập trực tiếp bộ nhớ video kết hợp chức năng dành riêng cho video, ví dụ như generator locking và đồng bộ hóa khung hình cho các hoạt động truy cập trực tiếp bộ nhớ khung được đồng bộ hóa hoàn toàn và chuyển qua truy cập trực tiếp bộ nhớ hình ảnh 2D. Ngoài việc đồng bộ hóa, số khung và thanh ghi hoạt động ở chế độ thu thập trực tiếp hoặc

phân tán để có thể dễ dàng kiểm soát bởi bộ xử lý trung tâm. Các thanh ghi khởi tạo, trạng thái và điều khiển trong lõi bộ truy cập bộ nhớ trực tiếp video được truy cập thông qua giao diện AXI4-Lite ở chế độ slave.

Bộ truy cập trực tiếp bộ nhớ video (VDMA) có thể được cấu hình để xử lý nhiều độ phân giải thông qua việc chỉ định độ phân giải ngang và dọc trong không gian thanh ghi của nó. Các giao diện Memory Mapper to Stream và Stream to Memory Mapper rộng 24 bit từ phiên bản AXI VDMA được kết nối với phiên bản AXI_MM (AXI Memory Mapped) của kết nối AXI. Các masters chạy theo xung nhịp video (bộ tạo xung nhịp trên bo mạch) yêu cầu bộ chuyển đổi xung nhịp không đồng bộ thành tần số lõi kết nối AXI 100 MHz. Khi mở rộng trong kết nối AXI được sử dụng để chuyển đổi các giao dịch 24 bit từ AXI VDMA sang độ rộng giao dịch của lõi kết nối AXI. Ngoài ra, bộ đệm dòng bên trong AXI VDMA cho các mặt đọc và ghi được đặt thành sâu 1K. Dữ liệu pixel từ giao diện truyền phát tạm thời được lưu trữ trong bộ đệm dòng trước khi nó được cung cấp trên giao diện MM. Điều này cho phép giao diện VDMA MM xử lý chuyển đổi độ rộng mà không mất dữ liệu.

Vi xử lí Microblaze ở chế độ bộ đệm lệnh và bộ đệm dữ liệu: Vi xử lý Microblaze chạy một số ứng dụng từ bộ nhớ chính, thiết lập và giám sát dữ liệu video vào ra.

Bộ tạo xung nhịp video là bộ tạo và phát hiện thời gian đa năng. Đầu vào của IP tự động phát hiện các xung nhịp đồng bộ hóa ngang và dọc, phân cực, khoảng trống, timing, và các điểm ảnh video đang hoạt động. Đầu ra của IP cũng tạo các xung nhịp đồng bộ được sử dụng trong các hệ thống video tiêu chuẩn và bao gồm hỗ trợ lập trình. Bộ tạo xung nhịp video chứa các giao diện AXI4-Lite để truy cập các thanh ghi slaves từ bộ xử lý.

Bộ cấu hình video đầu ra: Nó được sử dụng để chuyển đổi từ giao diện giao thức AXI4-Stream sang giao diện miền video. IP này hoạt động cùng với phần tạo thời gian của bộ tạo xung nhịp video. Nó cung cấp cầu nối giữa đầu vào video AXI4-Stream với đầu ra video (tín hiệu đồng bộ hóa video hoạt động với giao diện đồng bộ hóa, khoảng trống hoặc cả 2). Nó được cấu hình ở chế độ slaves vì nó nhận tín hiệu video trực tiếp từ khối VDMA và tín hiệu thời gian từ bộ tạo xung nhịp video để tạo tín hiệu đầu ra Video đến giao diện HDMI xuất ra màn hình.

Một bộ tạo xung nhịp và khối reset bộ xử lý cung cấp các xung nhịp và tín hiệu reset trên toàn hệ thống. Một bộ tạo xung nhịp khác trên bo mạch được sử dụng linh hoạt để thay

đổi tần số video trong thời gian chạy để thay đổi tốc độ khung hình. Hệ thống chứa các thiết bị ngoại vi I/O và IP hỗ trợ bộ xử lý được cung cấp bởi bộ xử lý MicroBlaze.

3.4 Thực hiện mô phỏng hệ thống đề xuất trên vivado

Sau khi đã thực hiện thiết kế hệ thống xử lý ảnh bằng các sơ đồ khối từ các bước đơn giản đến phức tạp thì học viên tiến hành thực nghiệm trên phần mềm Vivado để cho ra các kết quả kiểm chứng thực nghiệm và nhận xét những gì đã làm được khi phát triển hệ thống này.

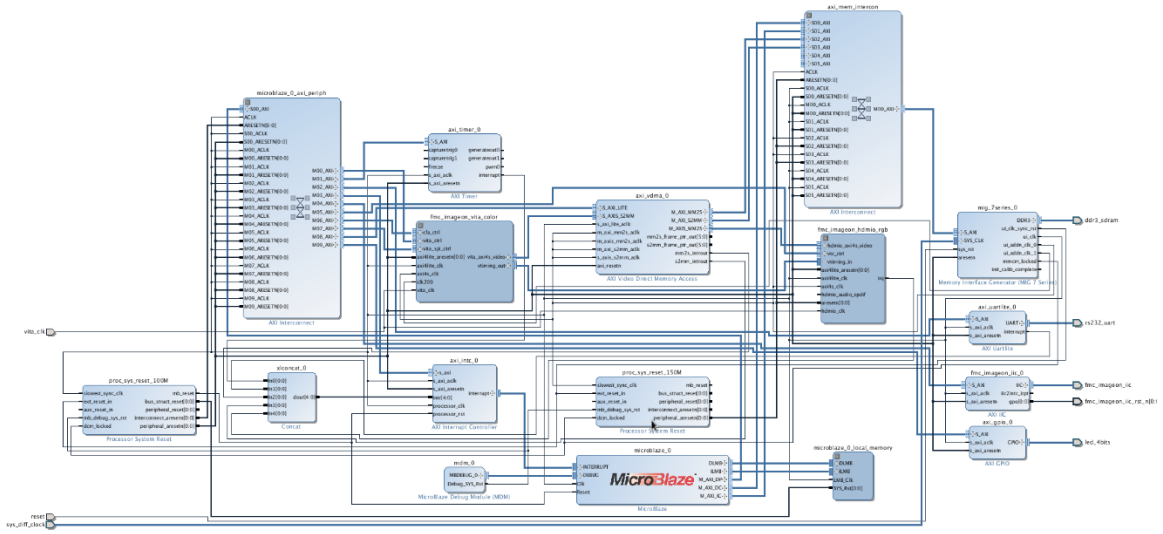
Đầu tiên tiến hành chạy thử một số ứng dụng cơ bản trên Artix-7 bằng project có sẵn trong vivado. Tiến hành chạy thử chương trình “Hello Word” và xuất kết quả ra màn hình LCD.

Hình 3.10: In chữ "Hello World" ra LCD trên kit Artix 7.

Kết quả đúng với mong muốn khi màn hình xuất hiện “Hello Word”. Sau đó học viên tiến hành tạo một project mới thực hiện ghép nối Camera bán dẫn Vita-2000 với Kit Artix-7. Kết quả thực nghiệm được kết nối camera với Kit không bị lỗi và đưa hình ảnh ra màn hình không bị mất dữ liệu.

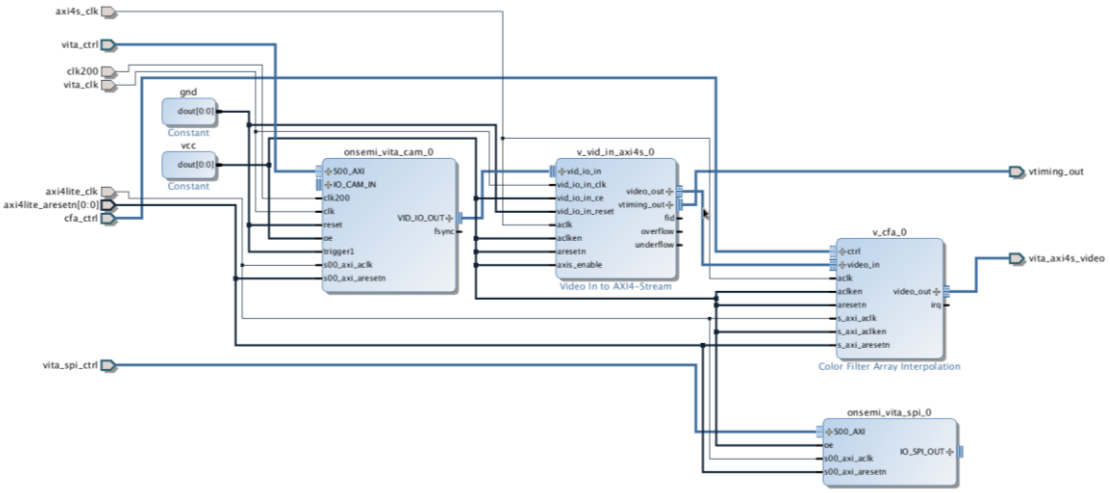
Hình 3.11: Ghép nối camera VITA-2000 và bo mạch FPGA Artix-7.

Thực hiện tạo và thiết kế các IP theo sơ đồ khối hệ thống đề xuất trên công cụ vivado. Thiết kế hệ thống được biểu diễn như hình 3.11:

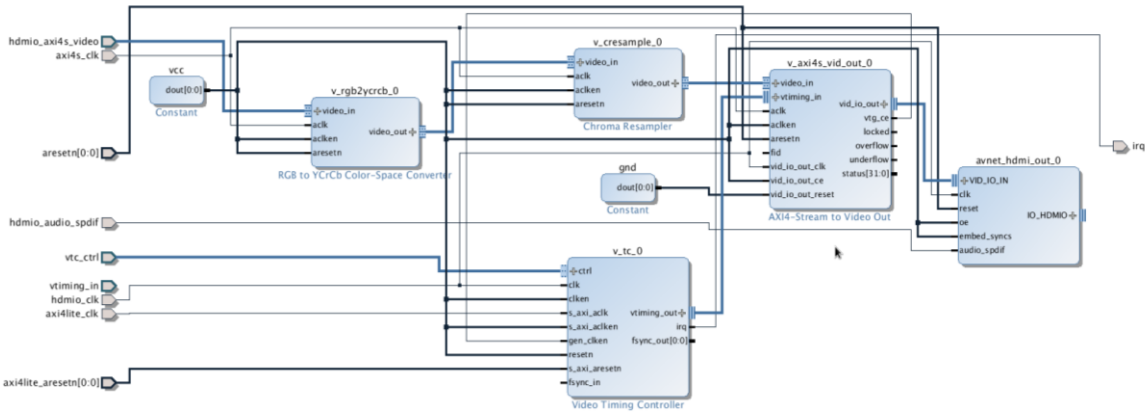


Hình 3.12: Hệ thống xử lý ảnh đề xuất trên vivado.

Trong đó, có hai IP trong hệ thống thực hiện việc kết nối camera và đưa dữ liệu ra công HDMI là IP fmc_imageon_vita_recever và IP fmc_imageon_hdmio_rgb của module Anvet FMC_IMAGEON các thành phần cụ thể được biểu diễn trong hình 3.13:

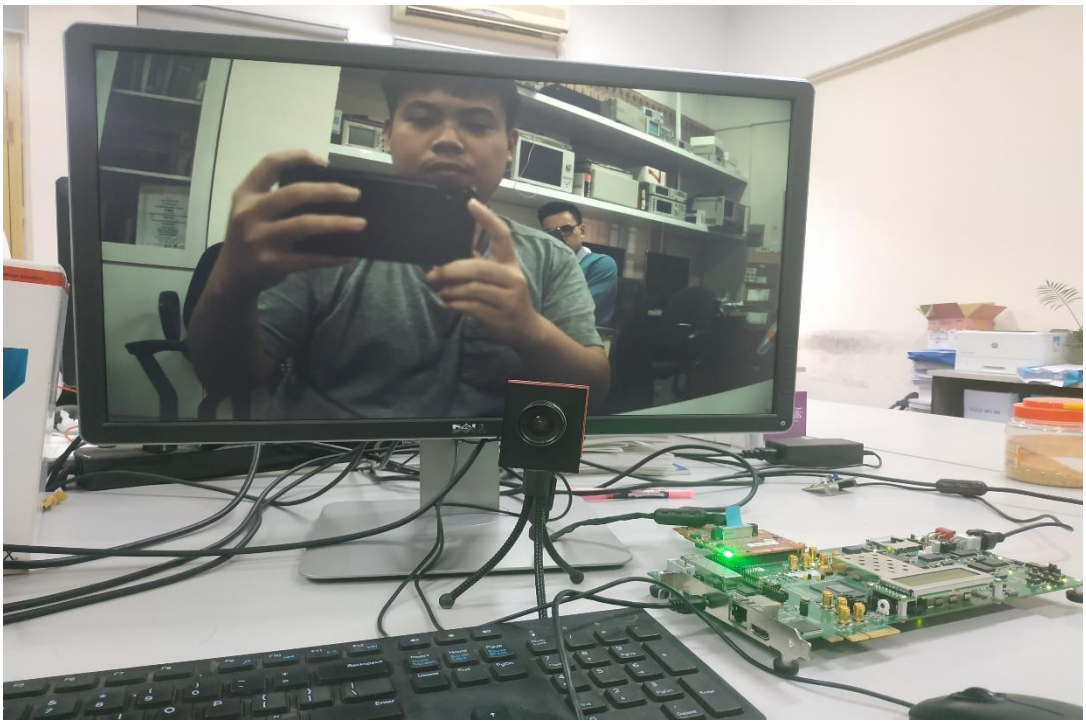


Hình 3.13: Kiến trúc khối IP fmc_imageon_vita_recever.



Hình 3.14: Kiến trúc khối IP fmc_imageon_hdmio_rgb.

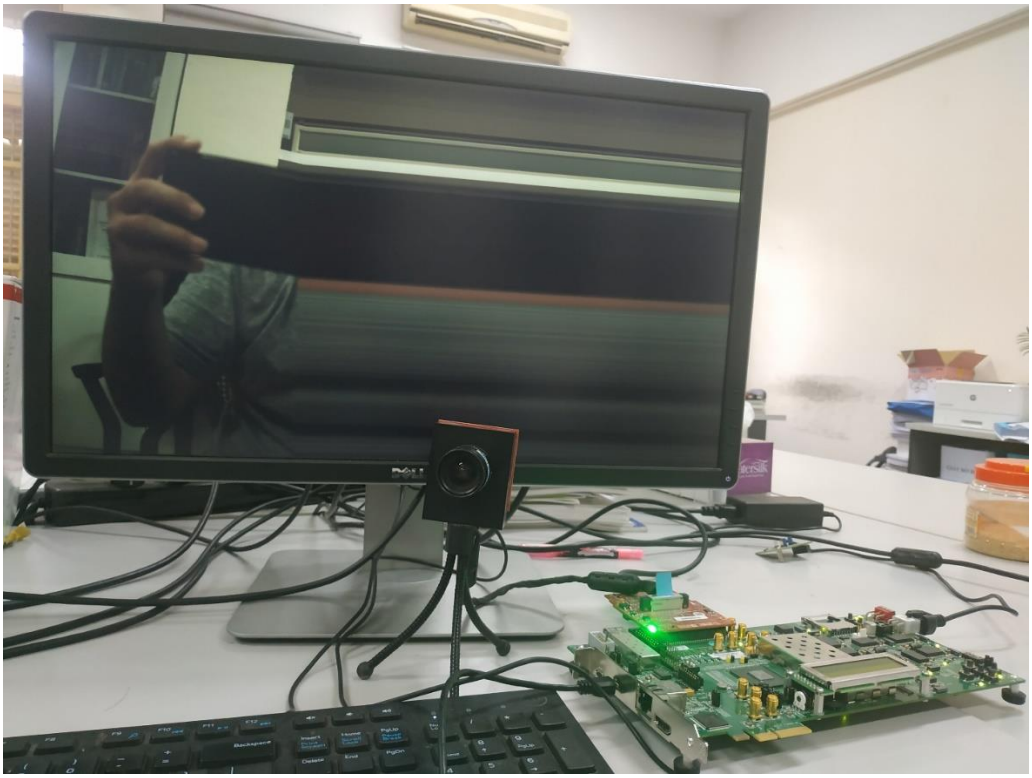
Sau khi tiến hành cấu hình cho các IP để thực hiện việc kết nối giữa Module Anvet FMC-IMAGEON và Artix-7 chúng ta tiến hành đưa kiến trúc phần cứng đã được thiết kế vào SDK để tiến hành mô phỏng và chạy thử đưa ra một số kết quả hiện thị hình ảnh trên màn hình. Đầu tiên trước khi kết nối với các IP xử lý ảnh đầu vào học viên sẽ tiến hành kết nối camera với màn hình thu được ảnh với chất lượng 1080p60 để xác nhận việc kết hợp camera với module hoàn thành và thu được kết quả như hình 3.15:



Hình 3.15: Hình ảnh camera VITA kết nối với bo mạch Artix-7

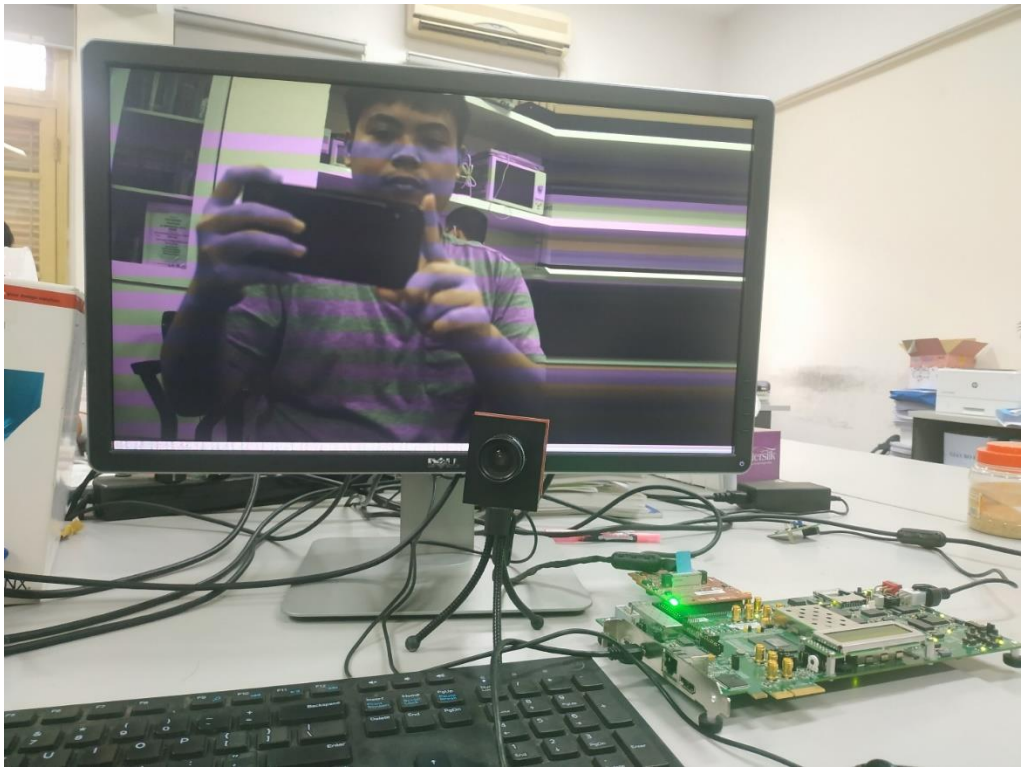
Sau đó học viên tiến hành kết nối IP VDMA vào trong hệ thống camera để tiến hành xử lý ảnh cũng như đưa ảnh vào vi xử lý để thực hiện các bước sau này. Sau khi thực hiện viết driver sử dụng khối VDMA học viên tiến hành thử nghiệm các trường hợp truyền tải hình ảnh quả khối VDMA để truyền ra ngoài, Tiến hành thử nghiệm khi cho độ phân giải của hình ảnh ở mức 1/3, 2/3 và full màn hình để tiến hành kiểm thử hệ thống có hoạt động được đúng như yêu cầu đã để ra.

Hình ảnh đầu ra bị một số lỗi như nhiễu đầu vào vẫn chưa xử lý được.



Hình 3.16: Cấu hình kích thước 1/3 khi đi qua khối VDMA.

Khi tiến hành cấu hình 1/3 màn hình thì hình ảnh đi qua có chất lượng tốt và không có nhiễu đầu ra. Sau đó học viên tiến hành thử nghiệm với đầu ra 2/3 màn hình thì cho ra kết quả với một số nhiễu trên khung hình. Lỗi ở đây có thể do sự đồng bộ hóa giữa bộ thu nhận video và khối VDMA chưa thật chính xác khiến cho các khung hình có thể bị nhiễu thậm chí là bị đè lên các khung hình khác.



Hình 3.17: Cấu hình kích thước 2/3 khi đi qua khối VDMA

Từ kết quả được thể hiện cụ thể trong bảng 3.3:

Bảng 3.3: Kết quả mô phỏng trên Vivado khi đi qua khối VDMA

Chiều ngang khung hình	0x780 (1920*1)	0xF00 (1920*2)	0x1680 (1920*3)
Nhiễu	không	có	Không tín hiệu
Kích thước video	1/3	2/3	Không tín hiệu

Sau khi đã chạy mô phỏng thực nghiệm trên bo mạch Artix-7 học viên tiến hành phân tích hệ thống phần cứng như về năng lượng hoạt động, hiệu năng cũng như tài nguyên sử dụng trong hệ thống đã thiết kế. Các số liệu sẽ được tổng hợp trong các bảng dưới đây:

Bảng 3.4: Bảng thống kê năng lượng và hiệu năng trong hệ thống

Thành phần trên Chip	Điện năng(W)	Sử dụng	Tổng	Tài nguyên sử dụng (%)
Xung nhịp	0,219	56	---	---
LUT as Lô-gic	0,067	32.957	134.600	24,49
Thanh ghi	0,006	46.617	269.200	17,32
LUT as Distributed RAM	0,004	4.021	46.200	8,70
CARRY4	0,004	1.454	33.650	4,32
Bộ phân kênh F7/F8	<0,001	987	134.600	0,73
LUT dạng thanh ghi dịch	<0,001	1.632	46.200	3,53
Khác	<0,001	2.352	---	---
Tín hiệu	0,126	76.546	---	---
RAM	0,052	38,5	365	10,55
MMCM	0,110	1	10	10,00
PLL	0,091	1	10	10,00
DSPs	0,016	12	740	1,62
I/O	1,119	168	400	42,00
PHASER	0,340	44	---	---
XADC	0,004	1	---	---
Năng lượng tĩnh	0,176	1	---	---
Tổng	2,332			

Như các số liệu thống kê trên bảng 3.4 về tổng hiệu năng cũng như năng lượng trên chip của các thành phần cơ bản, chúng ta có thể thấy được năng lượng tiêu hao nhiều nhất từ việc sản sinh ra các xung nhịp hay sử dụng các cổng I/O. Các xung nhịp dùng để cung cấp cho toàn bộ chip vì vậy số lượng xung nhịp phải sản sinh ra rất lớn chiếm tương đối nhiều năng lượng trong tổng năng lượng tiêu thụ. Tuy nhiên, thành phần chiếm gần 50% tổng năng lượng tiêu thụ của chip chính là các cổng I/O bởi vì việc truyền dữ liệu ra và vào từ camera với dung lượng lớn và liên tục, cho nên đòi hỏi nhiều năng lượng hơn để cung cấp cho nó. Việc cung cấp các tín hiệu trên chip cũng tương đối lớn, nguyên nhân là do việc truyền liên tục các tín hiệu điều khiển cũng như tín hiệu trạng thái về các khối để thực hiện việc truyền dữ liệu đến các khối khác. Ngoài ra còn có các bảng trạng thái LUT cũng

tiêu thụ một năng lượng không nhỏ. Tuy nhiên, chúng ta có thể thấy số lượng sử dụng các tín hiệu thì bo mạch chủ Artix-7 đủ đáp ứng nhu cầu để thực hiện việc thực thi hệ thống xử lý ảnh (hiệu năng cao nhất là các cổng vào ra I/O được sử dụng đến 42% trên tổng số cổng có trong bo mạch). Ngoài ra, chúng ta còn có được tổng năng lượng sử dụng từng phần trong các khối để chúng ta có thể thấy được khối nào sử dụng nhiều tài nguyên nhất trong chip.

Bảng 3.5: Điện năng sử dụng các khối trong hệ thống

Các khối trên Chip	Điện năng(W)	Tỉ lệ phần trăm (%)
HDMI_OUT	0,012	0,56
FMC_CAMERA_IN	0,087	4,03
MIG7	1,731	80,25
VDMA	0,031	1,44
FMC_CAMERA_SPI	0,005	0,23
MICROBLAZE	0,051	2,36
AXI_INTC	0,001	0,05
AXI_MEM_INTC	0,112	5,19
VTC	0,031	1,44
V_CFA	0,060	2,78
Các khối khác	0,036	1,67
Tổng	2,157	100

Có thể thấy được khối MIG7 là khối chiếm nhiều năng lượng sử dụng nhất bởi nó vừa thực hiện chức năng tạo các xung nhịp cung cấp cho các khối một chức năng quan trọng hơn là nó dùng để tạo giao diện kết nối tín hiệu với bộ nhớ DDR3 để sau này đưa tín hiệu đó đi xử lý. Vì vậy, chúng ta có thể thấy khối lượng thực thi tính toán cũng như khối lượng tín hiệu vào ra sử dụng trong khối MIG 7 là rất lớn cho nên nguồn tiêu thu của nó cũng lớn nhất là điều dễ hiểu. Tiếp đó chính là khối tiếp nhận tín hiệu camera đầu vào nguyên nhân vì tín hiệu đầu vào là video có dữ liệu vô cùng lớn cho nên năng lượng để đưa vào cũng lớn theo. Khối AXI_MEM_INTC chính là khối cung cấp giao diện để giao tiếp với bộ nhớ. Hệ thống đang thực thi luôn được lưu trữ dữ liệu trong bộ nhớ cho nên các tín hiệu cũng như dữ liệu đưa vào đây và năng lượng cũng cấp cũng lớn.

KẾT LUẬN

Hiện nay ngoài việc phát triển các thuật toán xử lý ảnh cũng như tốc độ xử lý thì việc thiết kế ra một hệ thống phần cứng, phần mềm đảm bảo được sự linh hoạt cũng như dễ dàng chỉnh sử trong một hệ thống điều đó góp phần giảm thiểu chi phí tái sử dụng cũng như có thể cứng hóa được các hệ thống xử lý ảnh hiện nay để đưa vào áp dụng cho các thuật toán xử lý ảnh sau này. Dựa vào nền tảng trong sự phát triển của các mạch bán dẫn cũng như các mạch tích hợp có thể tái cấu hình và hiệu năng cao. Trong khuôn khổ luận văn học viên đã thực hiện được ba công việc chính đó là: Tìm hiểu về hệ thống xử lý ảnh số các quá trình xử lý ảnh số; tìm hiểu về kiến trúc FPGA và công cụ cũng như các board FPGA được sử dụng hiện nay; đưa ra một hệ thống đề xuất tích hợp vào FPGA và được mô phỏng kiểm chứng kết quả bằng phần mềm.

Với kiến thức về xử lý ảnh số, học viên đã tìm hiểu được các bước trong xử lý ảnh số và các thành phần cấu thành nên một hệ thống xử lý ảnh số hiện này. Ngoài ra, còn có một số định nghĩa cũng như phương pháp trong xử lý ảnh số để làm gia tăng chất lượng ảnh đầu ra.

Với công nghệ FPGA, học viên đã tìm hiểu được tổng quan về kiến trúc các ưu điểm của FPGA từ đó so sánh với các mạch tích hợp khác để cho thấy sự phù hợp trong việc đưa vào đề tài. Học viên cũng tìm hiểu về bo mạch FPGA cụ thể là Artix-7.

Cuối cùng học viên đã đề xuất một hệ thống cơ bản trong việc xử lý ảnh bằng việc kết hợp camera VITA-2000 và bo mạch Artix-7 AC701 bằng công cụ mô phỏng Vivado. Hệ thống này mới chỉ là bước cơ bản tạo tiền đề để thiết kế ra các IP xử lý ảnh để đưa vào áp dụng cho các thuật toán sau này.

Hướng phát triển tiếp theo

Để có thể phát triển các ứng dụng thực tiễn, hệ thống cần tiếp tục được hoàn thiện, giải quyết triệt để các vấn đề liên quan đến nhiễu và việc truyền tải video. Trong tương lai học viên sẽ nghiên cứu xử lý được các nhiều đầu vào để không làm ảnh hưởng chất lượng video đầu ra và đưa dữ liệu đầu ra được toàn màn hình với chất lượng là 1080p.

Sau khi giải quyết được các vẫn đề trên, học viên sẽ tiến hành tạo một IP khác kết nối với hệ thống để tiến hành đưa thuật toán xử lý ảnh (theo dõi, phát hiện chuyện động) để hoàn thành xong đề tài và bước đầu có thể tạo ra một hệ thống phần cứng, phần mềm hoàn chỉnh.

TÀI LIỆU THAM KHẢO

[1] Xilinx, "AC701 Evaluation Board for the Artix-7 FPGA User Guide v1.4", August 6, 2019.

[2] Xilinx, "MicroBlaze Processor Reference Guide", June 21, 2018.

[3] Ian Kuon , Russell Tessier and Jonathan Rose, "FPGA Architecture: Survey and Challenges".

[4] Xilinx, "7 Series FPGAs Data Sheet: Overview", February 27, 2018.

[5] Rafael C. Gonzalez and Richard E. Woods. "Digital Image Processing".

[6] Anvet electronics marketing, "FMC-IMAGEON – VITA Pass-Through Tutorial verson 1.0".

[7] Xilinx, "AXI Video Direct Memory Access v6.2", November 30, 2016.

[8] Umer Farooq, Zied Marrakchi, Habib Mehres "Tree-based Heterogeneous FPGA Architectures: Application Specific Exploration and Optimization", 2012.

[9] Xilinx, "Video Timing Controller v6.1", May 22, 2019.

[10] A. Stalin and A. Wahi. BSFS, "Background subtraction frame difference algorithm for moving object detection and extraction", Journal of Theoretical and Applied Information Technology, February 28, 2014.

[11] Joana Barreto Martins, "Webcam motion detection and tracking interfaces for immersive embodied experiences", May, 2016.

[12] Nguyen Quang Hoan, "Xử lý ảnh", 2006.