

THIẾT KẾ SOPC DÙNG CHO CÁC ỨNG DỤNG XỬ LÝ ẢNH THỜI GIAN THỰC

Huỳnh Hữu Thuận, Đinh Quang Hải, Cao Trần Bảo Thương, Nguyễn Hữu Phương

Trường Đại Học Khoa Học Tự Nhiên, ĐHQG – HCM

(Bài nhận ngày 02 tháng 04 năm 2008, hoàn chỉnh sửa chữa ngày 06 tháng 06 năm 2008)

TÓM TẮT: Nội dung bài báo là thiết kế hệ thống trên một linh kiện logic khả trình (FPGA) dùng cho các ứng dụng xử lý ảnh thời gian thực. Để thực hiện điều này, chúng tôi thiết kế một số bộ gia tốc (Accelerator) để tăng tốc độ xử lý kết hợp với một số kỹ thuật xử lý đặc biệt (như truy xuất bộ nhớ trực tiếp – DMA, nhiều Master...). Kết quả được thử nghiệm trên DSP Development Kit sử dụng FPGA Stratix EP2S60F1020C4 của hãng Altera [1] dùng các phần mềm SoPC Builder, NIOS II IDE và Modelism làm môi trường kiểm tra kết quả và tích hợp phần cứng/phần mềm.

Từ khóa: FPGA, bộ gia tốc, DMA, SoPC, xử lý ảnh thời gian thực.

1. GIỚI THIỆU

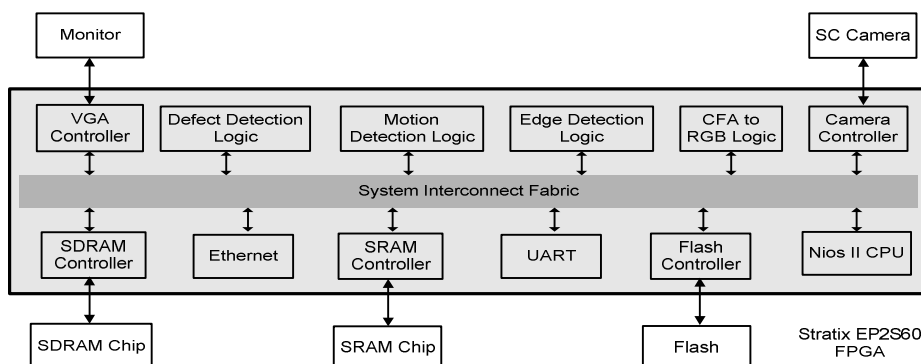
Trong những năm gần đây FPGA và các công cụ phát triển thiết kế SoPC (System On a Programmable Chip) trên FPGA đã cho thấy sự hiệu quả trong các ứng dụng xử lý dữ liệu tốc độ cao. Trong lĩnh vực xử lý ảnh đã có nhiều công trình nghiên cứu sử dụng SoPC cho các ứng dụng đòi hỏi độ phức tạp cao về mặt tính toán [2], [3]. Theo hướng này, chúng tôi đã phát triển một số thuật toán dùng cho xử lý ảnh trên phần cứng [4] – [6]. Nói chung, thiết kế một SoPC cho xử lý ảnh thời gian thực đòi hỏi nhiều khâu xử lý. Một vấn đề cần được giải quyết là thu nhận dữ liệu từ bên ngoài vào hệ thống với tốc độ cao và trao đổi dữ liệu với bộ nhớ ngoài (off – chip memory). Trong các Board mạch dùng để phát triển ứng dụng phổ biến thường chỉ có một (hoặc một vài) vi mạch nhớ được chia sẻ cho các ứng dụng khác nhau, dẫn đến lưu lượng trao đổi dữ liệu với bộ nhớ rất lớn, làm chậm hoạt động của toàn bộ hệ thống. Để giải quyết vấn đề này, chúng tôi dùng công cụ SoPC Builder của hãng Altera và khai thác kiến trúc System Interconnect Fabric (công nghệ tạo ra các kết nối của SoPC Builder dùng các nguồn tài nguyên logic bên trong FPGA để tích hợp các thành phần vào hệ thống) để phát triển các thiết kế. Ngoài các thành phần thường có trong một SoPC như CPU nhúng, bộ điều khiển SDRAM, bộ điều khiển SRAM, bộ điều khiển giao tiếp mạng, UART..., chúng tôi phát triển thêm các thành phần để giao tiếp với System Interconnect Fabric ở tốc độ cao dùng cơ chế truy xuất bộ nhớ trực tiếp DMA, cấu trúc nhiều Master, các mạch logic làm toán thay vì lập trình bằng phần mềm để điều khiển quá trình nhận dữ liệu từ Camera theo định dạng Bayer, chuyển sang dạng RGB dùng phương pháp được công bố trong [7] và GrayScale [8]. Chúng tôi thiết kế một số bộ gia tốc thực hiện các chức năng lọc, trừ ảnh... trên dữ liệu nhận được từ Camera để kiểm tra hoạt động của hệ thống.

2. TỔ CHỨC VÀ THIẾT KẾ HỆ THỐNG

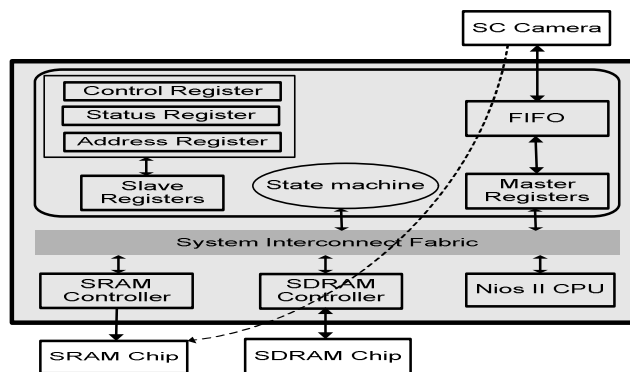
Sơ đồ khối tổng quát của hệ thống được trình bày trong Hình 1. Quá trình nhận dữ liệu từ Camera được điều khiển bởi khối Camera Controllor và lưu trữ vào SRAM ở dạng Bayer. Dữ liệu này sau đó được chuyển sang dạng RGB và GrayScale lưu trữ vào SDRAM. Việc sử dụng SRAM để lưu trữ dữ liệu từ Camera và SDRAM chứa dữ liệu RGB, GrayScale cho phép sử dụng hiệu quả kiến trúc nhiều Master của System Interconnect Fabric vì trong lúc một Master đọc dữ liệu từ SRAM để tính toán thì Master khác có thể ghi dữ liệu đã được chuyển đổi vào

SDRAM. Một bộ dò cạnh (Edge Detection) được sử dụng để tìm các cạnh trong ảnh dùng cho các ứng dụng xử lý ảnh.

Hình 2 trình bày quá trình lấy dữ liệu từ SC Camera bằng một cặp Master/ Slave. Cổng Slave cho phép truy xuất đến Status Register, Control Register và Address Register. Cổng Master chịu trách nhiệm truyền dữ liệu ảnh theo định dạng Bayer ngang qua System Interconnect Fabric. Để tăng hiệu quả của quá trình truyền dữ liệu, một FIFO được dùng để đệm dữ liệu từ Camera trước khi vào SRAM. Bộ nhớ SDRAM được dùng để lưu trữ lệnh cho hoạt động của CPU NIOS II nhúng. Nhiệm vụ của NIOS II trong trường hợp này là ghi vào Address Register địa chỉ bắt đầu của vùng nhớ lưu trữ dữ liệu đọc được từ Camera, sau đó khởi động quá trình truyền dữ liệu bằng cách cấu hình cho Control Register, và có thể kiểm tra trạng thái hoạt động của Camera Controller (như trạng thái của FIFO, ngắt...) bằng cách đọc Status Register.



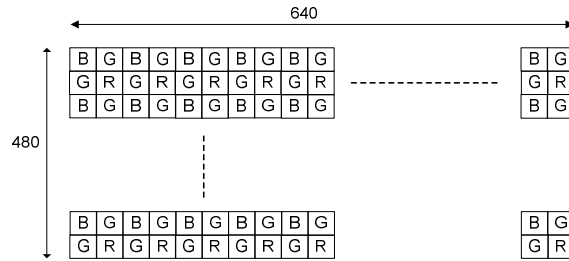
Hình 1. Sơ đồ khối tổng quát của hệ thống thiết kế.



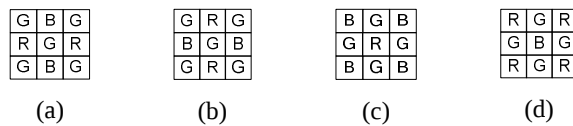
Hình 2. Quá trình lấy dữ liệu từ Camera vào bộ nhớ SRAM.

Dữ liệu nhận từ Camera ở dạng Bayer được minh họa trong Hình 3. Để chuyển sang dạng RGB, chúng ta cần nội suy hai giá trị màu bị mất tại mỗi Pixel. Một số thuật toán nội suy đã được công bố trong [8]. Để thuận tiện cho việc thực hiện trên phần cứng, chúng tôi sử dụng phương pháp nội suy tuyến tính ở Hình 4. Khi nội suy các giá trị của R và B ở vị trí của G, như ở các Hình 4 (a) và Hình 4 (b), trị giá trung bình của hai Pixel gần nhất được sử dụng. Ví

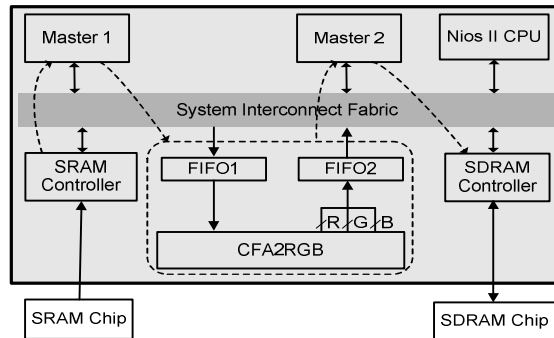
dụ trong Hình 4 (a), thành phần B tại Pixel G được tính bằng trị trung bình của hai Pixel B nằm trên và dưới Pixel G này, tương tự giá trị của thành phần R được tính bằng trị trung bình của hai Pixel R nằm bên trái và bên phải Pixel G. Hình 4 (c) trình bày trường hợp nội suy thành phần B tại Pixel R. Trong trường hợp đó, giá trị được nội suy là trị trung bình của bốn Pixel B ở góc và cách tính toán tương tự được áp dụng khi nội suy thành phần R tại Pixel B trong Hình 4 (d). Trong Hình 4 (c) và Hình 4 (d), trị giá G được nội suy bằng cách tính trung bình của bốn thành phần G gần nhất.



Hình 3: Dữ liệu nhận từ Camera.



Hình 4: Bốn trường hợp khi nội suy mẫu.

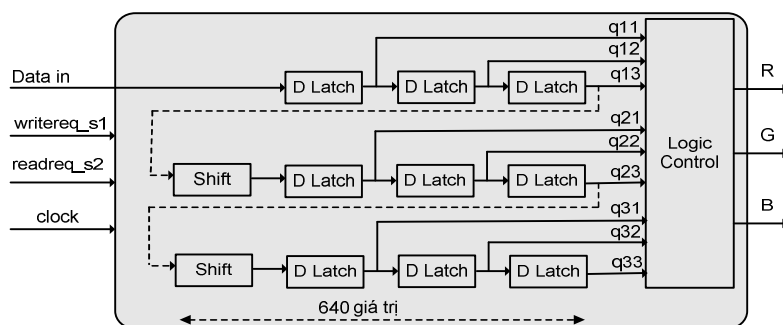


Hình 5: Kiến trúc thực hiện quá trình chuyển đổi từ dạng Bayer sang RGB.

Hình 5 trình bày thiết kế chuyển dữ liệu từ dạng Bayer được lưu trữ trong SRAM sang RGB lưu vào SDRAM. Master 1 thực hiện quá trình DMA với địa chỉ nguồn là SRAM và địa chỉ đích là FIFO1. Dữ liệu sau khi chuyển đổi được lưu vào SDRAM cũng bằng cơ chế DMA điều khiển bởi Master 2 với địa chỉ nguồn lúc này là FIFO2. Kiến trúc như vậy cho phép hai quá trình DMA được thực hiện đồng thời nên tốc độ chuyển đổi dữ liệu rất cao.

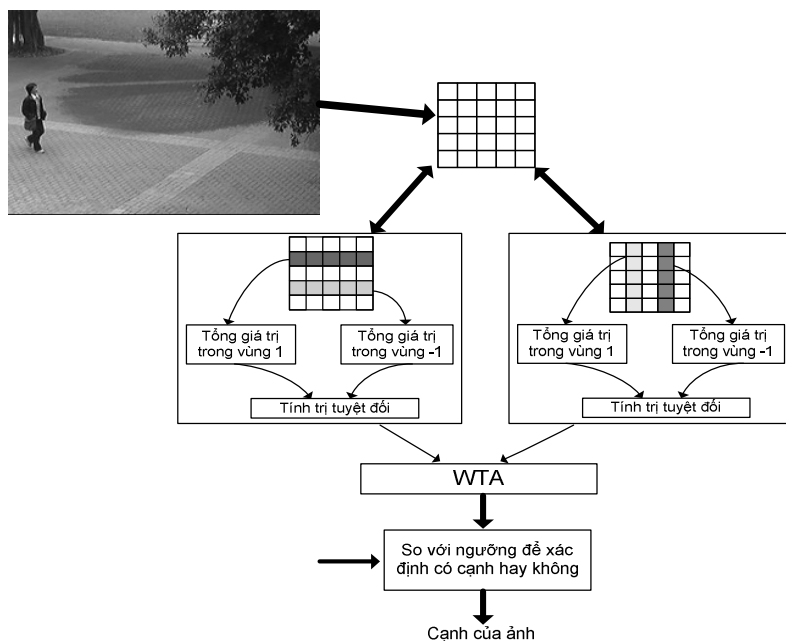
Cấu trúc của khối tính toán chuyển từ dạng Bayer sang RGB được trình bày trong Hình 6. Dữ liệu ngõ vào (Data in) được ghi lấy ra từ FIFO1 được đưa vào một dãy các thanh ghi dịch chuyển, tại mỗi lần dịch chuyển, 9 Pixel ở 3 hàng được lấy ra dùng cho quá trình nội suy mẫu. Cấu trúc này cho phép nội suy mỗi trị giá Pixel sau mỗi lần ghi dữ liệu nên rất hiệu quả trong quá trình xử lý các bộ lọc trên dữ liệu ảnh.

Kiến trúc còn có thể áp dụng hiệu quả cho các bộ lọc khác, ví dụ như thuật toán dò cạnh của ảnh cho bởi [9]. Cạnh chứa nhiều thông tin của ảnh và được dùng nhiều trong các ứng dụng xử lý ảnh như dò tìm chuyển động [6], nhận dạng [9]. Phương pháp thực hiện được minh họa ở Hình 7. Một cửa sổ 5x5 Pixel quét qua toàn bộ ảnh để dò tìm cạnh theo hướng dọc và hướng ngang. Quá trình này được thực hiện bằng cách tính tổng trong vùng 1 và -1 sau đó đưa qua mạch WTA (Winner Take All) [4], [5] để tìm ra hướng có khả năng là cạnh nhất. Giá trị này được quyết định là cạnh hay không tùy thuộc vào một trị giá ngưỡng (một trong những phương pháp xác định ngưỡng phù hợp với thiết kế phần cứng được trình bày trong [9]). Thiết kế như vậy rất phù hợp với phần cứng vì chỉ dùng các phép tính toán đơn giản. Kiến trúc thực hiện công việc này giống như đã đề xuất ở Hình 5 và Hình 6 nhưng sẽ dùng cửa sổ 5x5 Pixel thay vì 3x3 Pixel.

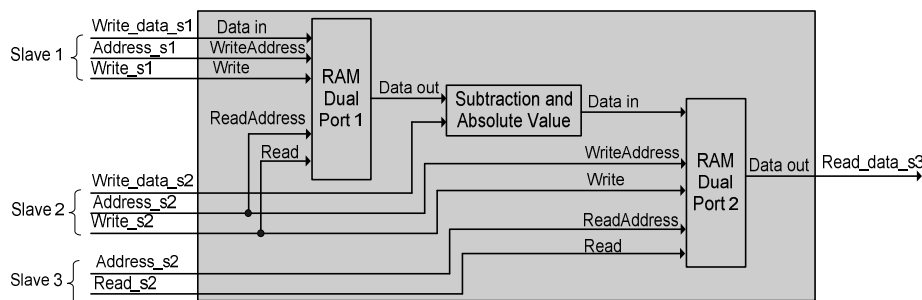


Hình 6: Khối thực hiện quá trình chuyển đổi từ dạng Bayer sang RGB.

Một thiết kế thường xuyên được sử dụng trong các ứng dụng dò tìm lỗi (defect detection), hay dò tìm chuyển động (motion detection) là trừ hai ảnh với nhau được minh họa ở Hình 8. Cấu trúc của Accelerator bao gồm ba Slave: Slave 1 dùng để ghi dữ liệu vào một bộ nhớ đệm thứ nhất (RAMDualPort1) có cấu trúc Dual Port cho quá trình phép đọc/ ghi dữ liệu đồng thời. Slave 2 dùng để ghi dữ liệu vào bộ nhớ đệm thứ hai (RAMDualPort2) cũng có cấu trúc Dual Port. Lưu ý là dữ liệu sẽ được trừ với dữ liệu trước đó đã được lưu trữ trong RAMDualPort1 và tính giá trị tuyệt đối trước khi ghi vào RAMDualPort2. Điều này được thực hiện bằng cách đọc dữ liệu từ RAMDualPort1 ra đồng thời với việc ghi dữ liệu vào RAMDualPort1 có nghĩa là địa chỉ đọc của RAMDualPort1 sẽ bằng với địa chỉ ghi của RAMDualPort2 và tín hiệu cho phép đọc dữ liệu từ RAMDualPort1 sẽ là tín hiệu ghi dữ liệu vào RAMDualPort2. Quá trình thực hiện trừ hai ảnh được thực hiện như sau: Đầu tiên, dữ liệu của ảnh thứ nhất được DMA vào Slave 1 và lưu trữ trong RAMDualPort1. Sau đó, dữ liệu của ảnh thứ hai được DMA vào RAMDualPort2. Kết quả là dữ liệu cần tính toán nằm trong RAMDualPort2 này và đọc ra bởi Slave 3.



Hình 7: Thuật toán dò tìm cạnh tương thích với phần cứng.

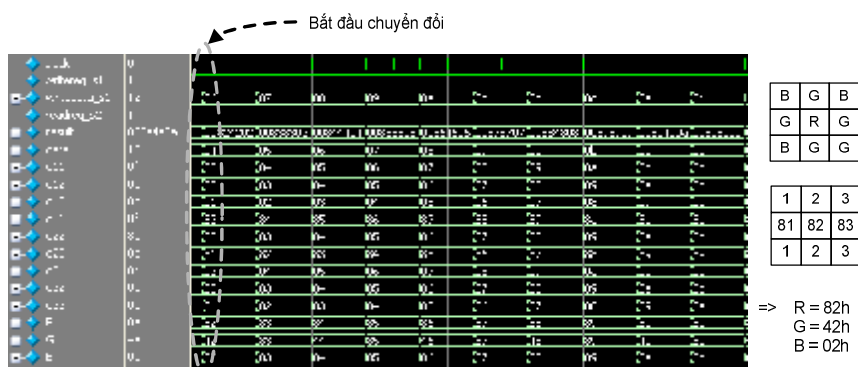


Hình 8: Kiến trúc của khối thực hiện phép trừ hai ảnh.

3.KẾT QUẢ THỰC NGHIỆM

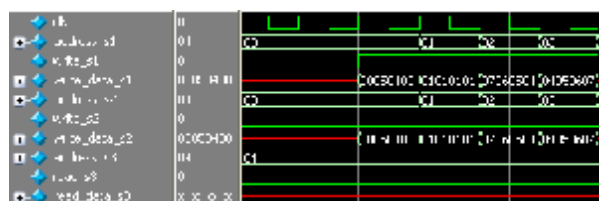
Quá trình thực nghiệm được tiến hành theo hai bước là mô phỏng và tổng hợp mạch để nạp xuống FPGA. Hình 9 trình bày kết quả thử nghiệm trên SoPC, NIOS II IDE và Modelsim cho thiết kế trình bày trong Hình 5. Để đạt hiệu quả truyền dữ liệu cao nhất thì mã lệnh, dữ liệu và ngăn xếp được lưu trong bộ nhớ bên trong FPGA (on-chip memory) tạo điều kiện cho Master 1 ghi dữ liệu vào FIFO1 từ SRAM (điều khiển bởi writereq_s1) và Master 2 đọc dữ liệu ra từ FIFO2 rồi ghi vào SDRAM (điều khiển bởi tín hiệu readreq_s2) với tốc độ 1 Pixel/clock và hai cặp Master/ Slave này chạy song song với nhau cho phép tốc độ chuyển đổi

có thể đạt đến 1 pixel/ clock. Kiến trúc cũng cho phép tích hợp khối logic thực hiện quá trình chuyển đổi sang dạng Grayscale từ dữ liệu RGB dùng công thức cho bởi [8].



Hình 9: Kết quả mô phỏng cho quá trình chuyển đổi từ dạng Bayer sang RGB dùng SoPC, Nios II IDE và ModelSim

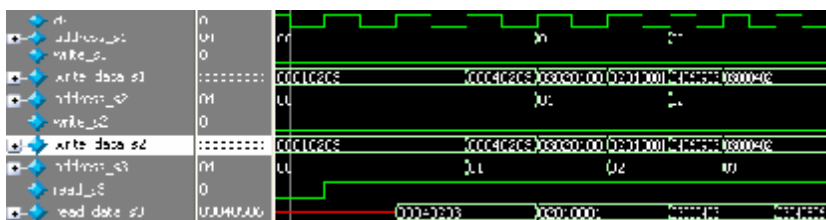
Hình 10, 11 và 12 trình bày kết quả mô phỏng của kiến trúc thực hiện trừ ảnh ở Hình 8 trong trường hợp thực hiện DMA 16 Pixel (4 Double Word). Hình 10 là kết quả khi thực hiện DMA 16 Pixel của ảnh thứ nhất vào Slave 1 (4 Double Word) ở Clock 50 Mhz. Dữ liệu lần lượt được ghi vào là 00050400H, 01010101H, 07060504H, 04050607H. Kết quả cho thấy dữ liệu được ghi vào với tốc độ rất cao (1 Double Word/Clock). Tương tự, Hình 11 trình bày quá trình DMA 16 Pixel của ảnh thứ hai vào Slave 2 cũng với tốc độ 1 Double Word/ Clock với các giá trị lần lượt là 00010203H, 03020100H, 04060902H, 01010101H. Hình 12 là kết quả đọc ra từ Slave 3 với tốc độ 4 Double Word/ 8 clock với kết quả lần lượt là: 00040203H, 02010001H, 03000402H, 03040506H.



Hình 10: Kết quả khi thực hiện DMA ghi dữ liệu vào Slave 1.



Hình 11: Kết quả khi thực hiện DMA ghi dữ liệu vào Slave 2.



Hình 12: Kết quả khi thực hiện DMA đọc dữ liệu từ Slave 3.

Sau khi kiểm tra thử nghiệm các thiết kế bằng SoPC Builder, Nios IDE và Modelsim, hệ thống được tổng hợp và tải xuống thử nghiệm trên SC Camera và DSP Development Kit dùng FPGA EP2S60 của hãng Altera như trình bày trong Hình 13 và 14. Kết quả tổng hợp mạch được trình bày trong Hình 15. Hình 16 (a) – (d) trình bày kết quả dùng cho dò tìm lỗi giữa hai Board mạch in bằng phương pháp trừ ảnh đơn giản. Trong Hình 16 (c), những vị trí có khác biệt được biểu diễn bằng giá trị 1 (thể hiện bằng màu trắng) và không khác biệt được biểu diễn bằng giá trị 0 (màu đen). Do ảnh chụp từ Camera bị nhiễu gây ra bởi môi trường nên ảnh khác biệt sau khi trừ nhau phải qua bộ lọc Median để quyết định kết quả cuối cùng.



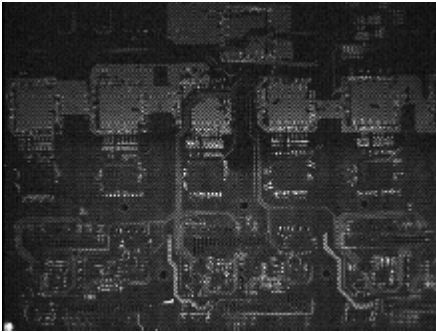
Hình 13: Camera thử nghiệm.



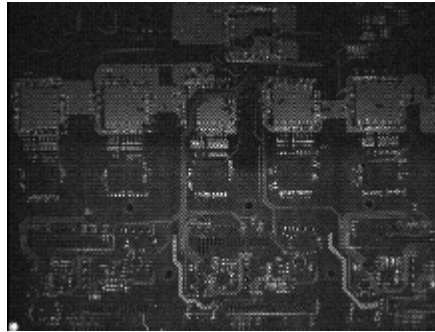
Hình 14: DSP Development Kit, Stratix Version.

Flow Status	Successful - Sat Feb 16 10:42:15 2008
Quartus II Version	7.2 Build 151 09/26/2007 SJ Full Version
Revision Name	standard
Top-level Entity Name	standard
Family	Stratix II
Device	EP2S60F1020C3
Timing Models	Final
Met timing requirements	No
Logic utilization	12 %
Combinational ALUTs	4,968 / 48,352 (10 %)
Dedicated logic registers	3,658 / 48,352 (8 %)
Total registers	3897
Total pins	205 / 719 (29 %)
Total virtual pins	0
Total block memory bits	120,544 / 2,544,192 (5 %)
DSP block 9-bit elements	8 / 288 (3 %)
Total PLLs	1 / 12 (8 %)
Total DLLs	0 / 2 (0 %)

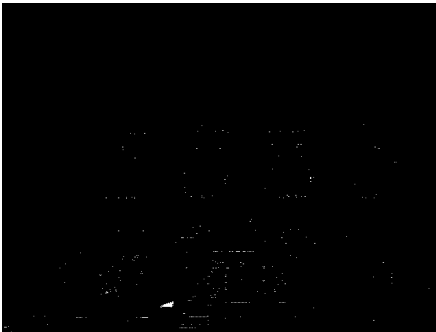
Hình 15: Kết quả tổng hợp mạch.



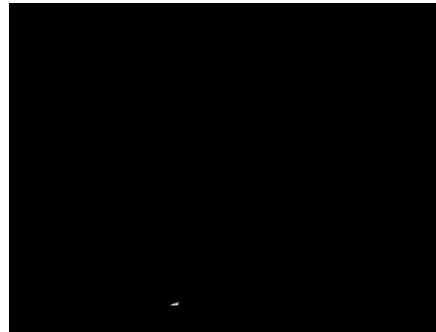
Hình 16 (a): Ảnh GrayScale thứ 1.



Hình 16 (b): Ảnh GrayScale thứ 2.



Hình 16 (c): Ảnh sau khi trừ nhau.



Hình 16 (d): Ảnh sau khi qua lọc Median.

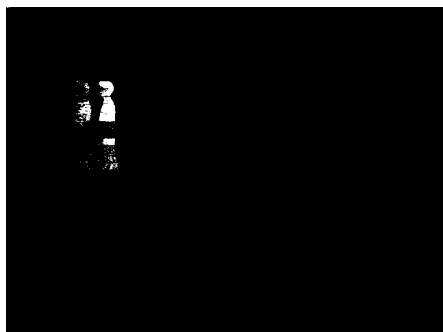
Trường hợp dò tìm chuyển động bằng phương pháp trừ ảnh và dùng ngưỡng đơn giản cũng được thử nghiệm với kết quả được trình bày trong các Hình 17 (a) – (c). Hình 17 (c) cho thấy vị trí của đối tượng chuyển động.



Hình 17 (a): Ảnh GrayScale ở thời điểm T1.



Hình 17 (b): Ảnh GrayScale ở thời điểm T2.



Hình 17 (c): Ảnh sau khi trừ nhau kết hợp với ngưỡng.

4.KẾT LUẬN

Chúng tôi đã thiết kế một hệ thống trên chip FPGA dùng cho các ứng dụng xử lý ảnh thời gian thực. Một số kỹ thuật đặc biệt đã được sử dụng nhằm tăng tốc độ của hệ thống (DMA, thực hiện việc tính toán bằng phần cứng, sử dụng cấu trúc nhiều Master/ Slave để thiết kế các bộ gia tốc) dựa trên cấu trúc System Interconnect Fabric của SoPC Builder, hãng Altera. Các thuật toán được sử dụng tương thích với thiết kế phần cứng và phù hợp với thiết kế trên FPGA. Một số kết quả thử nghiệm được minh họa kiểm chứng trên SoPC Builder, Nios II IDE, Modelsim và cuối cùng nạp xuống FPGA của DSP Development Kit để kiểm tra. Các kết quả cho thấy việc thiết kế là tốt.

DESIGNING A SOPC FOR REAL - TIME IMAGE PROCESSING APPLICATIONS

Huynh Huu Thuan, Dinh Quang Hai, Cao Tran Bao Thuong, Nguyen Huu Phuong
University of Natural Sciences, VNU – HCM

ABSTRACT: *This paper presents our design of a System On a Programmable Chip (SoPC) for real - time image processing applications. To do this, some accelerators are designed and integrated into a FPGA chip in order to process data at a high speed. Besides, special techniques (such as DMA, multiple Masters...) are also used to speed up the system. The result is tested on a DSP Development Kit using FPGA Stratix II EP2S60F1020C4 of Altera Corporation. SoPC Builder, NIOS II IDE and Modesim are used for testing and integrating hardware/software.*

Keywords: *FPGA, Accelerator, DMA, SoPC, Real-time image processing.*

TÀI LIỆU THAM KHẢO

- [1]. Tài liệu về DSP Development Kit, Stratix Edition ở Website: <http://www.altera.com/products/devkits/altera/kit-dsp-2S60.html>.
- [2]. R.-X. Chen, Jeffrey Fan, *Complexity reduction for SOPC-based H.264/AVC coder via sum of absolute difference*, IEEE/CIE 7th International Conference on ASIC (ASICON'07), pp. 1277-1280, Oct. (2007).
- [3]. Roman C. Kordasiewicz, Shahram Shirani, *On Hardware Implementations Of DCT and Quantization Blocks for H.264/AVC* VLSI Signal Processing, pp. 93-102, (2007).
- [4]. Huynh Huu Thuan, Le Duc Hung, *A VLSI Architecture For Manhattan Distance and WTA*, ISEE 2005, Symposium on Electrical & Electronics Engineering 2005 – Oct. 11, 12 2005 - HCM City, Vietnam.
- [5]. Huynh Huu Thuan, Cao Tran Bao Thuong, Nguyen Huu Phuong, *Design of a Flexible VLSI Architecture for Full-Search Vector Quantization based on Minimum Mean Square Criterion*, Tạp Chí Khoa Học Và Công Nghệ, Viện Khoa Học và Công Nghệ Quốc Gia, tập 44, số 3, (2006).
- [6]. Trần Ngọc Phụng, Trần Thị Diễm, Cao Trần Bảo Thương, Huỳnh Hữu Thuận, *Thực Hiện Một Số Thuật Toán Dò Tìm Chuyển Động Trên FPGA*, Hội Nghị Khoa Học Trường Đại Học KHTN lần 5, (2006).
- [7]. Tadashi Sakamoto, Chikako Nakanishi and Tomohiro Hase, *Software Pixel Intorpolation for digital still cameras suitable for a 32-bit MCU*, IEEE Transactions on Consumer Electronics, Vol. 44, No. 4, Nov. (1998).
- [8]. Carey Bunks, *Grokking the GIMP*, New Riders Publishing, www.newriders.com, (2000).
- [9]. M. Yagi and T. Shibata, *An Image Representation Algorithm Compatible With Neural-Associative – Processor - Based Hardware Recognition Systems*, IEEE transactions on neural networks, vol. 14, no.5, pp.1144-1161, Sep. (2003).