



THIẾT KẾ ỨNG DỤNG PAINT TRÊN NỀN FPGA

Trương Phong Tuyên¹, Tạ Duy Thắng¹, Lưu Tioni¹ và Lương Vinh Quốc Danh¹

¹ Khoa Công nghệ, Trường Đại học Cần Thơ

Thông tin chung:

Ngày nhận: 19/12/2014

Ngày chấp nhận: 17/08/2015

Title:

Implementation of an FPGA-based Paint application

Từ khóa:

Altera DE2-115, đồ họa máy tính, DE2-Paint, FPGA, hệ thống nhúng

Keywords:

Altera DE2-115, computer graphics, DE2-Paint, FPGA, embedded systems

ABSTRACT

Nowadays, designing embedded systems on FPGA technology has proven to be an effective solution for various applications because of its noteworthy benefits such as high speed, high performance and low power consumption. The project “Design and implementation of an FPGA-based Paint application”, therefore, aims to create an embedded application on FPGA to perform some fundamental computer graphics algorithms. Importantly, further research into computer graphics and embedded systems on FPGA will be started out based on these results.

TÓM TẮT

Hiện nay, các hệ thống nhúng trên nền FPGA với nhiều ưu điểm vượt trội như tốc độ, hiệu suất xử lý cao và tiêu hao năng lượng thấp đang nhận được sự quan tâm nghiên cứu và ứng dụng ngày càng tăng. Nghiên cứu “Thiết kế ứng dụng Paint trên nền FPGA” được thực hiện với mục tiêu thi công một ứng dụng nhúng sử dụng FPGA thực hiện một số giải thuật đồ họa máy tính cơ bản. Những kết quả của nghiên cứu này sẽ là cơ sở cho việc tiếp tục phát triển các ứng dụng nhúng về đồ họa và giải trí đa phương tiện trên nền FPGA.

1 GIỚI THIỆU

Việc phát triển các hệ thống nhúng cho các ứng dụng đồ họa máy tính không còn là lĩnh vực quá mới mẻ với một số quốc gia phát triển. Nhiều công ty, trường đại học nổi tiếng trên thế giới đã có những nghiên cứu nguyên cứu thành công, cho ra đời nhiều sản phẩm nổi tiếng trong lĩnh vực này. Tuy nhiên tại Việt Nam, chúng ta chỉ mới từng bước tiếp cận với lĩnh vực này khoảng vài năm trở lại đây.

Từ thực tế trên, mục tiêu của nghiên cứu “Thiết kế ứng dụng Paint trên nền FPGA” được thực hiện trên board DE2-115 Development and Education của Terasic nhằm bước đầu nghiên cứu và phát triển một hệ thống nhúng phục vụ các ứng dụng đồ họa máy tính. Qua đó góp phần bổ sung vào nguồn tư liệu tham khảo cho các nghiên cứu và phát triển ứng dụng nhúng tiếp theo của sinh viên chuyên

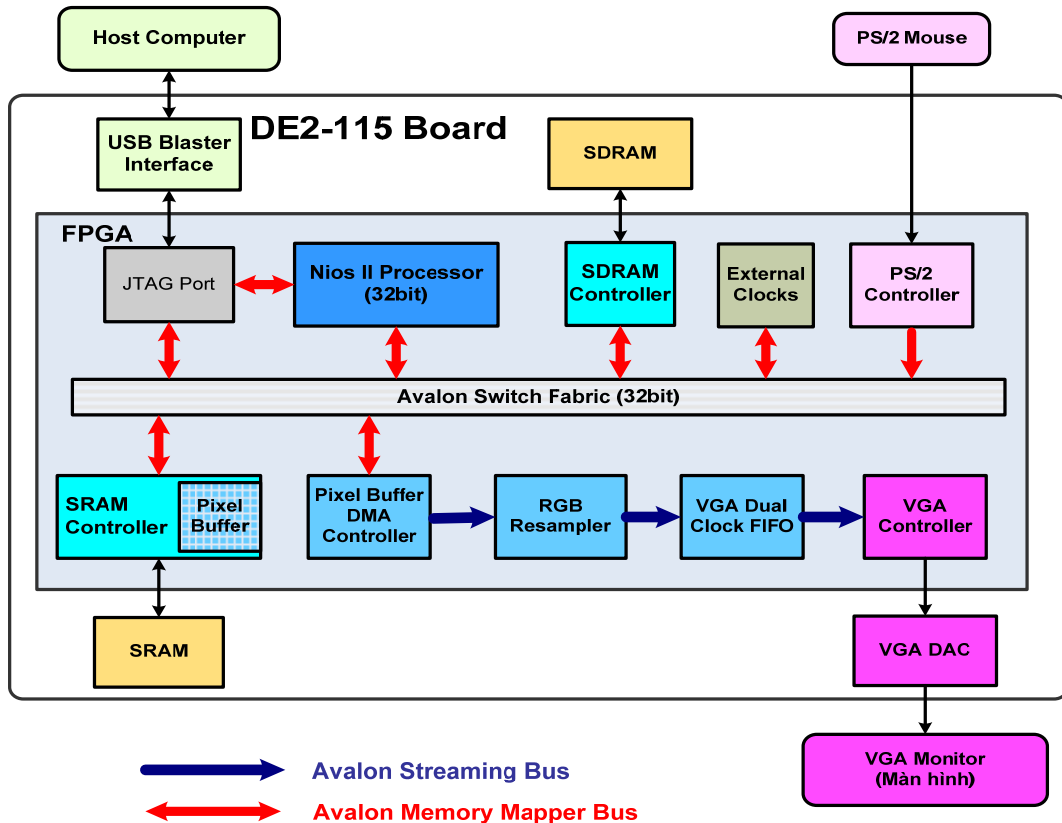
ngành Kỹ thuật Máy tính. Trong thời gian khoảng 4 tháng thực hiện, nghiên cứu đã tập trung vào việc phát triển một hệ thống nhúng có khả năng thực hiện các giải thuật đồ họa cơ bản như vẽ và tô màu đoạn thẳng, đa giác, hình tròn,... Các giải thuật đồ họa đã được cài đặt, minh họa dưới giao diện chương trình tương tự như ứng dụng Microsoft Paint trên hệ điều hành Windows.

2 PHƯƠNG PHÁP THỰC HIỆN

2.1 Tổng quan về hệ thống

2.1.1 Phần cứng

Hệ thống được thiết kế và cấu hình trên board DE2-115 của Terasic bằng cách sử dụng bộ công cụ SOPC Builder trong phần mềm Quartus II của Altera. Hệ thống có khả năng giao tiếp với chuột qua cổng PS/2 cho phép xử lý và hiển thị hình ảnh lên màn hình (Terasic Technologies Inc, 2013).



Hình 1: Sơ đồ khối hệ thống

Các thành phần chính của hệ thống:

- *Nios II processor*: khối xử lý trung tâm đảm nhiệm xử lý các lệnh thực thi trên hệ thống (tần số hoạt động 50MHz). Vì xử lý Nios II sử dụng SDRAM trên board DE2-115 làm bộ nhớ dữ liệu truy xuất thông qua lõi điều khiển SDRAM.

- *Avalon Switch Fabric*: bus (32-bit) có nhiệm vụ liên kết các thành phần của hệ thống.

- *Pixel Buffer*: sử dụng một phần vùng nhớ SRAM (tối đa 2MHz) để lưu trữ dữ liệu hình ảnh.

- *Các khối xử lý hình ảnh (Pixel Buffer DMA Controller, RGB Reasample, VGA Dual Clock FIFO)*: thực hiện các công việc đọc hình ảnh từ bộ nhớ, chuyển đổi định dạng, không gian màu của hình ảnh.

- *Khối VGA Controller*: Thực hiện giao tiếp với module VGA-DAC để hiển thị hình ảnh lên màn hình.

- *P/S2 Controller*: Thực hiện giao tiếp với thiết bị PS/2 (chuột/ bàn phím).

Để hiển thị hình ảnh lên màn hình VGA, hình

ảnh cần phải được nạp vào bộ nhớ Pixel Buffer. Khi đó các thành phần hệ thống sẽ thực hiện việc chuyển đổi định dạng hình ảnh và hiển thị ra màn hình nhờ bộ điều khiển VGA (Adam Chapweske, 2003), (Altera Corporation, 2010).

2.1.2 Phần mềm

Phần mềm được thiết kế, biên dịch và cài đặt dựa trên bộ công cụ Nios II trong gói phần mềm Altera Quartus II (Altera Corporation, 2007). Về chức năng, phần mềm bao gồm các giải thuật chính:

- Giao tiếp với màn hình và chuột thông qua chuẩn VGA và PS/2.

- Thao tác trên các pixel (xóa pixel, thiết lập/lấy giá trị pixel, ...)

- Vẽ đoạn thẳng và hình học (hình tròn/ elip, các đa giác, ...)

- Tô màu, xử lý, hiển thị ký tự, ...

2.2 Thiết kế hệ thống nhúng

2.2.1 Thiết kế hệ thống trên phần mềm SoPC Builder

Conn...	Module	Description	Clock	Base	End	IRQ
	CPU	Nios II Processor	[clk]			
	instruction_master	Avalon Memory Mapped Master	sys_clk			
	data_master	Avalon Memory Mapped Master	[clk]			
	jtag_debug_module	Avalon Memory Mapped Slave	[clk]			
	SDRAM	SDRAM Controller	sys_clk	0x0a000000	0x0a0007ff	IRQ 0
	Pixel_Buffer	SRAM/SSRAM Controller	sys_clk	0x08000000	0x081fffff	IRQ 31
	HEX3_HEX0	Parallel Port	sys_clk	0x10000020	0x1000002f	
	HEX7_HEX4	Parallel Port	sys_clk	0x10000030	0x1000003f	
	PS2_Port	PS2 Controller	sys_clk	0x10000100	0x10000107	
	JTAG_UART	JTAG UART	sys_clk	0x10001000	0x10001007	
	Interval_Timer	Interval Timer	sys_clk	0x10002000	0x1000201f	
	sysid	System ID Peripheral	sys_clk	0x10002020	0x10002027	
	External_Clocks	Clocks Signals for DE-Series Board Pe...	clk	0x10002030	0x10002031	
	Pixel_Buffer_DMA_C...	Pixel Buffer DMA Controller	sys_clk	0x10003020	0x1000302f	
	RGB_Resampler	RGB Resampler	sys_clk			
	VGA_Dual_Clock_FIFO	Dual-Clock FIFO	multiple			
	VGA_Controller	VGA Controller	vga_clk			

Hình 2: Hệ thống nhúng được thiết kế

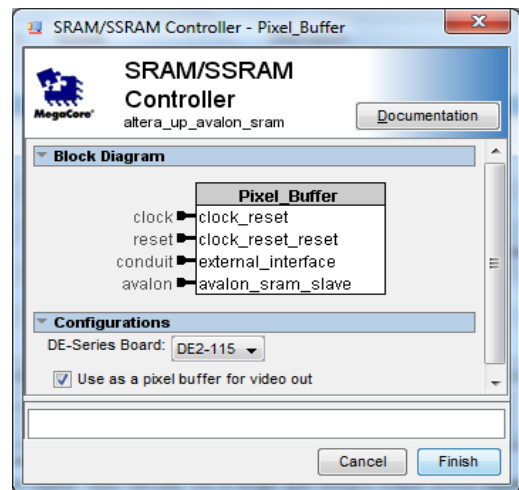
Hình 2 là kết quả thiết kế phần cứng của hệ thống sử dụng công cụ SOPC Builder được tích hợp trong gói phần mềm Quartus II của hãng Altera. Các thành phần của hệ thống trên được Altera cung cấp hầu hết dưới dạng các lõi IP (Intellectual Property core). Để xây dựng hệ thống, người thiết kế cần tiến hành lựa chọn và cấu hình các lõi IP một cách hợp lý đáp ứng các yêu cầu đặt ra như sau:

Nios II processor (CPU)

Nios II processor là một lõi (core) vi xử lý 32 bit được tối ưu cho các ứng dụng của hãng sản xuất Altera. Bộ xử lý này đảm nhiệm xử lý các lệnh thực thi, điều khiển hoạt động của các thành phần phần cứng trong hệ thống. Trong đề tài này, hệ thống SoPC sử dụng phiên bản NiosII/s (phiên bản chuẩn), phiên bản này yêu cầu tài nguyên thấp trên chip FPGA mà vẫn đảm bảo đáp ứng tốt các yêu cầu đặt ra (Altera Corporation, 2011).

Pixel Buffer

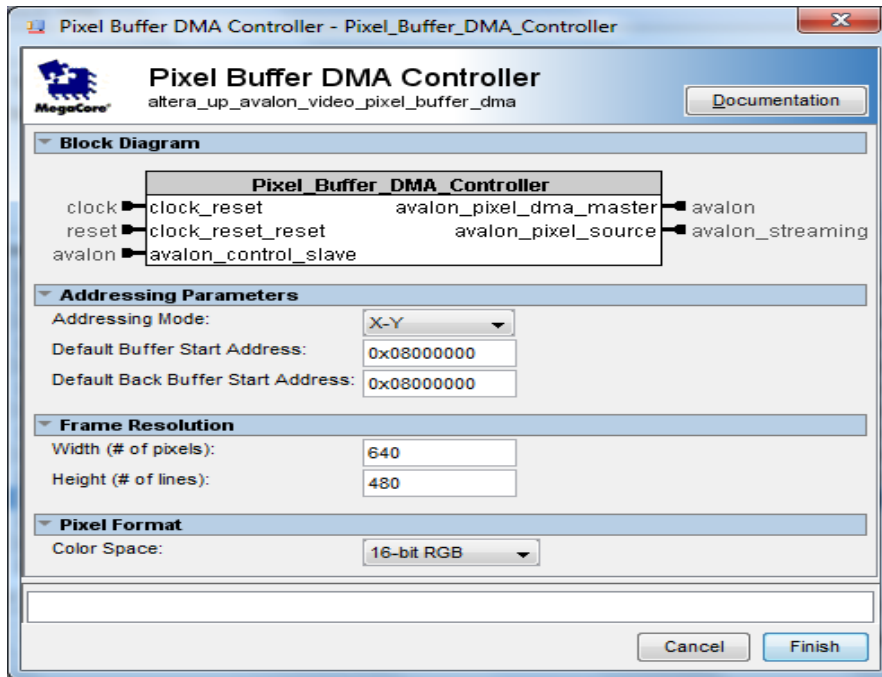
Khối Pixel Buffer sử dụng một phần vùng nhớ SRAM trên board DE2-115 để lưu trữ các giá trị pixel của hình ảnh (có độ phân giải 640x480 pixel, định dạng 16bit-RGB). Trong hệ thống SoPC, người thiết kế cần cấu hình cho lõi SRAM/SSRAM Controller hoạt động như một bộ Pixel Buffer tương thích với board DE2-115 như ở Hình 3. Khối Pixel Buffer trong hệ thống có địa chỉ từ 0x08000000 đến 0x081fffff (Altera Corporation, 2010).



Hình 3: Cấu hình khối Pixel Buffer

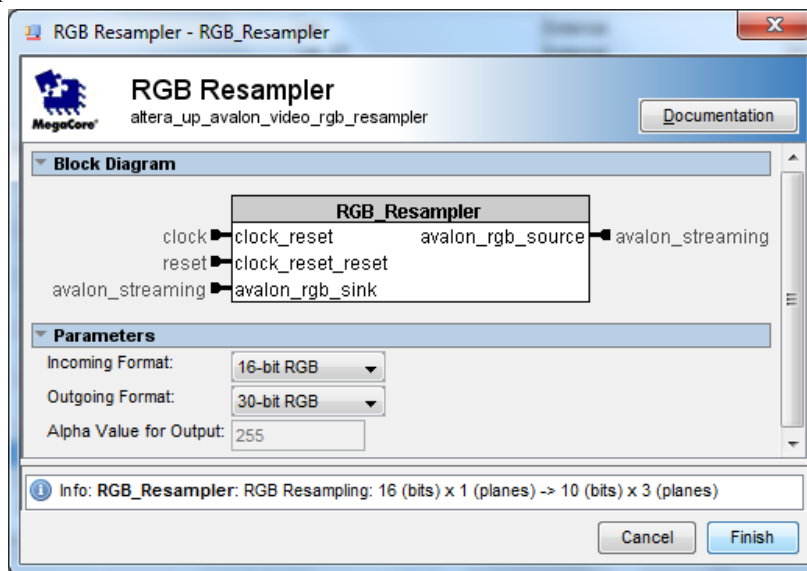
Pixel Buffer DMA Controller

Khối Pixel Buffer DMA Controller được xem như là khối được sử dụng để ánh xạ dữ liệu từ bộ nhớ Pixel Buffer để làm dữ liệu đầu vào cho các khối xử lý tiếp theo. Để khối Pixel Buffer DMA Controller có thể đọc chính xác dữ liệu hình ảnh lưu trong khối Pixel buffer, cần thiết lập các thông số về độ phân giải của hình ảnh (640x480 pixel), định dạng không gian màu (16 bit-RGB), địa chỉ bắt đầu đọc dữ liệu 0x08000000 tương ứng với dữ liệu hình ảnh lưu trên khối Pixel Buffer (Hình 4) (Altera Corporation, 2010).



Hình 4: Cấu hình khối Pixel Buffer DMA Controller

VGA Resampler



Hình 5: Cấu hình khối RGB Resampler

Dữ liệu hình ảnh lưu trong bộ nhớ Pixel Buffer có độ phân giải 640x480 và được định dạng 16-bit-RGB. Vì vậy, cần phải sử dụng khối VGA Resampler để chuyển đổi định dạng ngõ vào (incoming format) 16-bit RGB thành 30-bit RGB ở ngõ ra (outgoing format). Định dạng này thích hợp với định dạng dữ liệu đầu vào của khối VGA

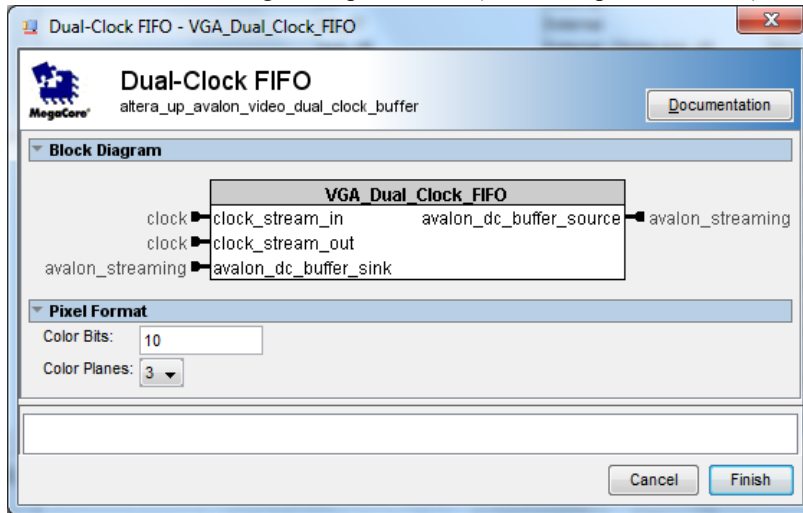
Controller, 640x480 pixel, 30-bit RGB như Hình 5 (Altera Corporation, 2010).

VGA Dual Clock FIFO

Hình 6 là hình ảnh thiết lập các thông số của VGA Dual Clock FIFO. Đây là một bộ đệm dữ liệu và giúp đồng bộ cho quá trình vận chuyển dữ liệu

giữa thành phần có 2 nguồn xung clock khác nhau. Trong hệ thống, khối VGA Controller sử dụng xung clock 25 MHz, trong khi các thành phần khác của hệ thống hoạt động ở xung clock 50MHz. Khối VGA Dual Clock FIFO đảm bảo việc giao tiếp

giữa khối VGA Resample và khối VGA Controller được đồng bộ với nhau. Trong hệ thống này khối VGA Dual Clock FIFO được định dạng ở không gian màu của hình ảnh 30 bit RGB, 10 bit cho mỗi màu (Altera Corporation, 2010).



Hình 6: Cấu hình khối Dual Clock (DC) FIFO

VGA Controller

Lỗi VGA Controller cung cấp dữ liệu hình ảnh, đồng thời tạo ra các tín hiệu định thời cho module VGA-DAC trên board DE2-115. Ngoài ra, VGA Controller cũng tạo ra cả tín hiệu đồng bộ ngang và dọc sử dụng cho việc quét hình ảnh trên màn hình. Lỗi VGA Controller cung cấp bởi nhà sản xuất có thể tạo ra tín hiệu định thời ở tần số 25MHz (Altera Corporation, 2010).

PS/2 Controller

Lỗi PS/2 quản lý sự định thời của giao thức truyền dữ liệu nối tiếp PS/2 (PS/2 Serial Data Transmission Protocol). Trình điều khiển thiết bị có thể giao tiếp với bộ phận này bằng cách đọc và ghi dữ liệu cũng như điều khiển các thanh ghi. Lỗi PS/2 Controller sử dụng trong hệ thống với 256-word FIFO để lưu trữ dữ liệu nhận được từ thiết bị PS/2 (bàn phím/ chuột). Địa chỉ của lỗi điều khiển PS/2 trong hệ thống từ 0x10000100 đến 0x10000107 và được dùng để giao tiếp với chuột PS/2 để thực hiện các thao tác vẽ ảnh (Adam Chapweske, 2003; Altium Corporation, 2009; Andries Brouwer, 2009).

2.2.2 Phần mềm

Việc thực hiện các giải thuật đồ họa dù đơn giản hay phức tạp về cơ bản đều được thao tác trên từng pixel. Trong nghiên cứu này, nhóm tác giả sử dụng 02 giải thuật điển hình là giải thuật vẽ đoạn

thẳng và giải thuật tô màu (Donald Hearn, M. Pauline Baker, 1986; Lâm Thị Ngọc Châu, 2003; Lê Tấn Hùng, Huỳnh Quyết Thắng, 2000).

Giải thuật vẽ đoạn thẳng Bresenham

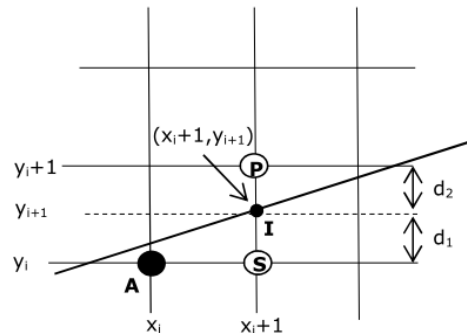
Xét phương trình đoạn thẳng:

$$y = kx + m \tag{1}$$

với $k = (y_2 - y_1) / (x_2 - x_1)$ và $m = y_1 - kx_1$

Trong đó k được gọi là độ dốc hay hệ số góc của đoạn thẳng và m được gọi là đoạn chắn trên trục y.

Từ phương trình có thể xây dựng quá trình vẽ các đoạn thẳng khi cho x biến thiên theo các khoảng Δx và kết quả ta có thể thu được giá trị của biến y thay đổi với các khoảng Δy tương ứng ($\Delta y = k\Delta x$).



Hình 7: Minh họa giải thuật Bresenham

Xét giải thuật Bresenham, giả sử bước thứ i , ta đã xác định được tọa độ (x_i, y_i) , như vậy chúng ta cần xác định tọa độ (x_{i+1}, y_{i+1}) cho bước kế tiếp. Với giá trị $x_{i+1} = x_i + 1$, giá trị của y_{i+1} có thể được chọn một trong hai giá trị y_i và $y_i + 1$. Cách chọn được mô tả như sau:

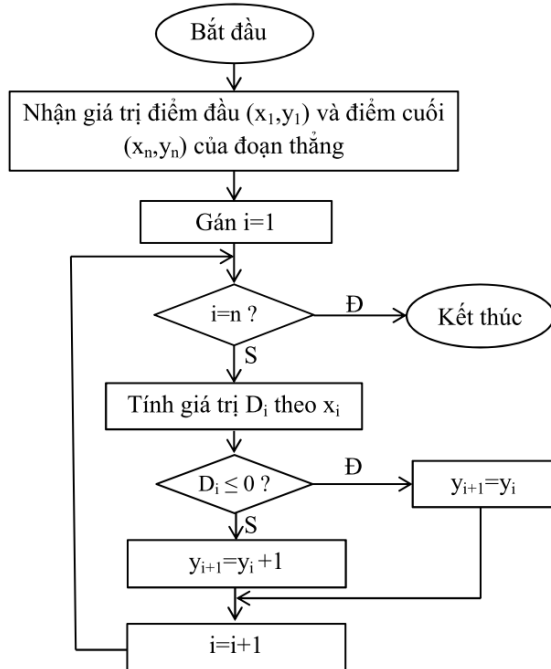
$$d_1 = y_{i+1} - y_i = k(x_i + 1) + m - y_i \quad (2)$$

$$d_2 = y_i + 1 - y_{i+1} = y_i + 1 - k(x_i + 1) - m \quad (3)$$

$$\text{Đặt } D_i = d_1 - d_2$$

Nếu: $D_i \leq 0 \leftrightarrow d_1 \leq d_2$ ta chọn S : $y_{i+1} = y_i$

$D_i > 0 \leftrightarrow d_1 > d_2$ ta chọn P : $y_{i+1} = y_i + 1$



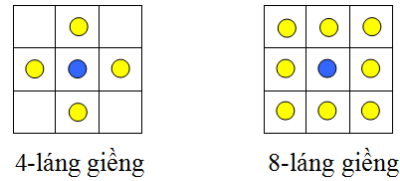
Hình 8: Lưu đồ giải thuật vẽ đoạn thẳng Bresenham

Trong lưu đồ Hình 8, một vòng lặp được sử dụng để duyệt theo trục x từ đầu đến cuối đoạn thẳng $(x_1 \rightarrow x_n)$. Với mỗi giá trị x_i ta tính giá trị D_i tương ứng, từ đó xác định giá trị của y_i . Việc này giúp ta chọn được các tọa độ (x_i, y_i) nguyên (trùng ứng với các pixel) gần với đường thẳng cần vẽ nhất.

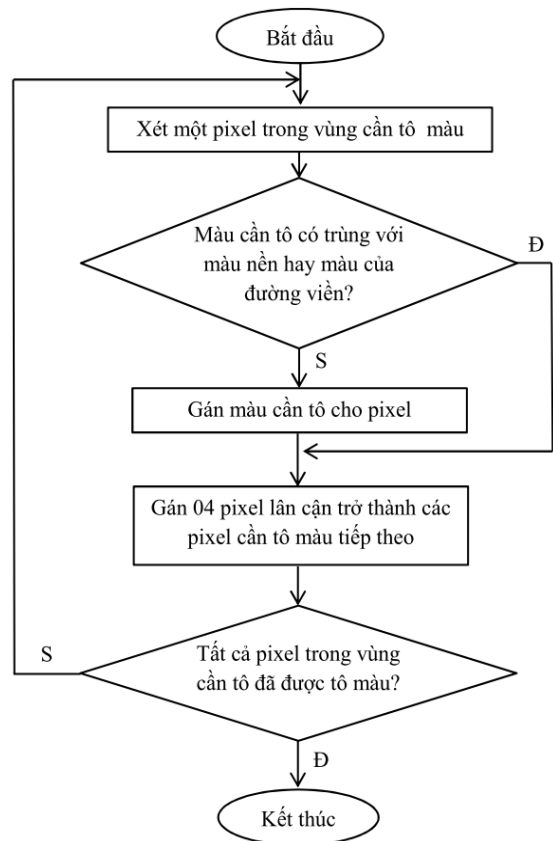
Giải thuật tô màu

Tô màu một vùng là thay đổi màu sắc của các điểm vẽ nằm trong vùng cần tô. Một vùng cần tô màu thường được xác định bởi một đường khép kín nào đó gọi là đường biên (viên). Hàm tô màu một vùng đa giác bất kỳ sử dụng giải thuật tô màu loang (Flood Fill). Giải thuật loang là một giải thuật đệ quy dựa trên ý tưởng: xuất phát từ một

điểm seed (điểm bắt đầu tô màu) duyệt qua 4 điểm hoặc 8 điểm láng giềng (điểm lân cận) của điểm seed. Nếu điểm chưa được tô thì sẽ tô màu điểm đó. Điểm này trở thành điểm seed mới. Quá trình lặp lại cho đến khi toàn vùng được tô màu.



Hình 9: Minh họa giải thuật tô màu loang



Hình 10: Lưu đồ giải thuật tô màu loang

Hình 10 trình bày lưu đồ giải thuật tô màu loang cho một vùng với 4 láng giềng. Bắt đầu từ 01 pixel trong vùng cần tô màu, giải thuật tô màu loang sẽ được áp dụng tô màu cho 04 pixel láng giềng. Việc tô màu trên sẽ lặp lại cho đến khi tất cả các pixel trong vùng cần tô đều được tô màu.

3 KẾT QUẢ VÀ ĐÁNH GIÁ

Nội dung nghiên cứu tập trung vào việc thiết kế và thực hiện một hệ thống nhúng trên FPGA có khả năng thực hiện và hiển thị một số giải thuật đồ

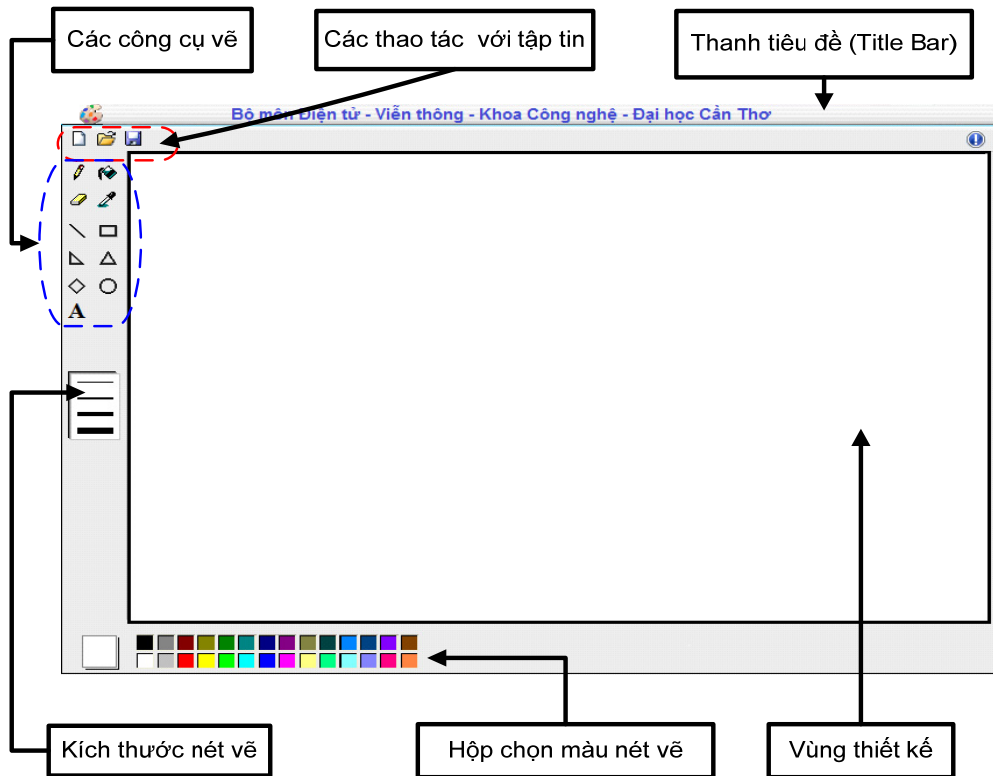
họa máy tính (vẽ đoạn thẳng, các đa giác, hình tròn, êlip, nhập ký tự, ...).

Về phần cứng, các thành phần trong hệ thống

và các ngoại vi được đồng bộ và hoạt động tốt. Toàn bộ hệ thống được cấu hình trên chip FPGA Cyclone IV EP4CE115F29 với số tài nguyên được sử dụng như Hình 11.

Flow Summary	
Flow Status	Successful - Sat Apr 20 10:47:20 2013
Quartus II Version	10.1 Build 153 11/29/2010 SJ Full Version
Revision Name	DE2_115_Media_Computer
Top-level Entity Name	DE2_115_Media_Computer
Family	Cyclone IV E
Device	EP4CE115F29C7
Timing Models	Final
▲ Total logic elements	11,886 / 114,480 (10 %)
Total combinational functions	10,521 / 114,480 (9 %)
Dedicated logic registers	7,375 / 114,480 (6 %)
Total registers	7649
Total pins	206 / 529 (39 %)
Total virtual pins	0
Total memory bits	310,076 / 3,981,312 (8 %)
Embedded Multiplier 9-bit elements	11 / 532 (2 %)
Total PLLs	1 / 4 (25 %)

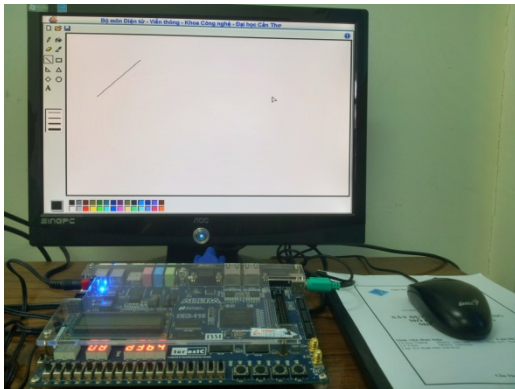
Hình 11: Tổng số tài nguyên hệ thống sử dụng trên chip FPGA



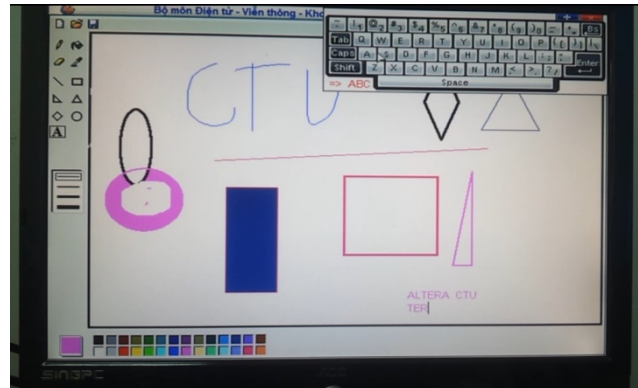
Hình 12: Giao diện chính của chương trình

Hệ thống sử dụng vi xử lý Nios II với tần số xung nhịp 50MHz, bộ nhớ SDRAM 128MB, giao tiếp với màn hình LCD và chuột máy tính. Hình

ảnh hiển thị trên màn hình có độ phân giải 640x480 pixel, định dạng không gian màu RGB 16-bit.



(a)



(b)

Hình 12: (a) Phần cứng thiết kế sử dụng board FPGA DE2 của hãng Altera; (b) Các thao tác với ứng dụng Paint chạy trên phần cứng được thiết kế

Phần mềm được lập trình bằng ngôn ngữ C với giao diện được thiết kế cho phép thực hiện được tất cả các thao tác vẽ, tô màu và lưu ảnh như với ứng dụng Microsoft Paint thường thấy trên hệ điều hành Microsoft Windows (Hình 12). Bên cạnh đó, hệ thống cũng bổ sung tính năng nhập bàn phím ảo cho phép nhập văn bản từ việc click chuột hoặc thông qua màn hình cảm ứng. Bố trí hệ thống trong thực nghiệm và hình ảnh minh họa một số thao tác trên hệ thống như Hình 12. Một đoạn video clip trình bày kết quả của nghiên cứu có thể xem tại đường link <http://goo.gl/MRkyHZ>.

4 KẾT LUẬN VÀ ĐỀ XUẤT

Nghiên cứu đã cơ bản đạt được các yêu cầu đặt ra về thiết kế một phần cứng nhúng trên nền FPGA cho phép thực hiện các giải thuật đồ họa cơ bản. Các giải thuật đồ họa này đã được minh họa thông qua một ứng dụng vẽ và tô màu đơn giản tương tự như ứng dụng Paint của Microsoft. Mặc dù vậy, trong quá trình thực hiện do khả năng xử lý của hệ thống còn hạn chế nên màu sắc của hình ảnh hiển thị chưa thật sống động, đôi lúc còn bị trễ. Tuy nhiên, với những kết quả đạt được nêu trên cho thấy một hướng nghiên cứu đúng đắn và đây sẽ là nguồn tham khảo hữu ích cho sinh viên ngành Kỹ thuật máy tính và Công nghệ thông tin trong lĩnh vực thiết kế hệ thống nhúng cho các ứng dụng đồ họa máy tính.

TÀI LIỆU THAM KHẢO

1. Adam Chapweske, 2003. The PS/2 Mouse Interface. URL: <http://www.computer-engineering.org/>, ngày truy cập 15/09/2014.
2. Altera Corporation, 2007. Quartus II Handbook Volume 4: SOPC Bulder. URL:

www.cs.columbia.edu/~sedwards/classes/2008/4840/qts_qii5v4.pdf, ngày truy cập 15/09/2014.

3. Altera Corporation, 2010. Media Computer System for the Altera DE2-115 Board. URL: ftp://ftp.altera.com/up/pub/Altera.../DE2-115_Media_Computer.pdf, ngày truy cập 15/09/2014.
4. Altera Corporation, 2010. Video IP Cores for Altera DE Series Boards. URL: ftp://ftp.altera.com/up/pub/Altera_Material/9.1/University_Program_IP_Cores/Audio_Video/Video.pdf, ngày truy cập 15/09/2014.
5. Altera Corporation, 2011. Nios II Processor Reference Handbook. URL: www.altera.com/literature/hb/nios2/n2cpu_nii5v1.pdf, ngày truy cập 15/09/2014.
6. Altium Corporation, 2009. PS2 Commands. <http://techdocs.altium.com/display/FPGA/PS2+Commands>, ngày truy cập 15/09/2014.
7. Andries Brouwer, 2009. Keyboard scancodes. <http://www.win.tue.nl/~aeb/linux/kbd/scancodes-13.html>, ngày truy cập 15/09/2014.
8. Donald G. Bailey, 2011. Design for Embedded Image Processing on FPGAs. John Wiley & Sons Pte Ltd. 352 pp.
9. Donald Hearn, M. Pauline Baker, 1986. Computer Graphics. Prentice-Hall, Inc. Englewood Cliffs. New Jersey. 352 pp.
10. Hồ Nguyễn Đạt, 2009. Hệ thống nhúng: hướng đi mới của Việt Nam. Thời báo Văn minh Sài Gòn, http://tbvtsg.com.vn/show_article.php?id=16049&ln_id=101, ngày truy cập 15/09/2014.

11. Lâm Thị Ngọc Châu, 2003. Kỹ thuật đồ họa. Khoa CNTT&TT. Trường Đại học Cần Thơ. 159 trang.
12. Lê Tấn Hùng, Huỳnh Quyết Thắng, 2000. Kỹ thuật đồ họa. Nhà xuất bản Khoa học và Kỹ thuật. Hà Nội. 355 trang.
13. MathWorks, Inc, 2014. MatLab – Image Processing Toolbox 6 User’s Guide. http://www.mathworks.com/help/pdf_doc/images/images_tb.pdf, ngày truy cập 15/09/2014.
14. Phạm Thượng Cát, 2005. Hệ thống nhúng – Thời kỳ hậu máy tính. Thế giới máy tính <http://www.pcworld.com.vn/articles/kinh-doanh/giai-phap/2005/06/1187654/he-thong-nhung-thoi-ky-hau-may-tinh/>, ngày truy cập 15/09/2014.
15. Ranjani Chandrasekar, Manu Jain, 2007. FPGA Paint, Advanced Microcontrollers Final Projects. http://people.ece.cornell.edu/land/courses/ece5760/FinalProjects/f2007/rc437_mj288/rc437_mj288/rc437_mj288/index.html, ngày truy cập 15/09/2014.
16. Terasic Technologies Inc, 2013. DE2-115 User Manual. http://www.terasic.com.tw/cgi-bin/page/archive_download.pl?Language=English&No=502&FID=cd9c7c1feaa2467c58c9aa4cc02131af, ngày truy cập 15/09/2014.