

Nghiên cứu và thiết kế ứng dụng hệ thống truyền thông giữa HPS-FPGA trên nền SoC

Research and Application design of a communication system between HPS-FPGA based on SoCs

Đỗ Văn Cần, Nguyễn Phùng Quang, Đoàn Quang Vinh
ĐH Quy Nhơn, ĐH Bách Khoa Hà Nội, ĐH Đà Nẵng
Email: dovancan@qu.edu.vn

Tóm tắt

Dòng SoC FPGA (SoC) được cấu thành bởi hai thành phần đó là HPS và FPGA. Trong đó, phần thứ nhất chứa lõi xử lý cứng ARM, cho phép thực thi những chương trình ứng dụng phức tạp. Phần thứ hai cho phép người dùng tùy biến cấu hình và thực hiện nhiều thiết kế IP trên thành phần này.

Từ khi SoC ra đời, một vài phương thức truyền thông được xây dựng để thực hiện giao tiếp giữa HPS và FPGA ví dụ như Axi Bridge, Avalon MM Pipeline Bridge...

Nhóm tác giả nghiên cứu, thiết kế ứng dụng hệ thống truyền thông giữa HPS và FPGA trên SoC của Altera bằng công cụ Qsys nhằm mang lại hiệu quả cao trong việc trao đổi dữ liệu cho CNC-on-Chip.

Từ khóa: IP, truyền thông HPS-FPGA, truyền thông SoCs, CNC-on-Chip, Axi Bridge.

Abstract

The SoC FPGA device is a single-die system on a chip (SoC) that consists of two distinct parts - a HPS portion and a FPGA portion. In which, the first part contains a hard-core ARM processor that allows the implementation of complex applications. The second part allows user customizable configuration and perform many IP designs for this component.

Since SoCs are produced, a few modes of communication have been developed to make communicating between the HPS and FPGA eg: Axi Bridge, Avalon MM, Pipeline Bridge...

The authors' research, application design of a communication system between HPS and FPGA base Altera's SoCs FPGA, using Qsys tool to deliver effective data exchange for CNC-on-Chip.

Keywords: IP, communication HPS-FPGA, communication SoCs, CNC-on-Chip, Axi Bridge

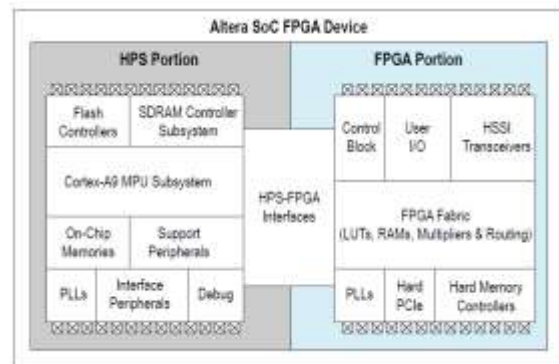
Chữ viết tắt

AXI	Advanced eXtensible Interface
CNC	Computer Numerical Control
DSP	Digital Signal Processor
FPGA	Field Programmable Gate Array
GPV	Global Programmers View
HDL	Hardware Description Language
HPS	Hard Processor System
IP	Intellectual Property
SoC	System on Chip

1. Phần mở đầu

Dòng SoC FPGA (sau đây gọi tắt là SoC) được cấu thành từ hai thành phần đó là HPS và FPGA như trên hình H.1. Phần thứ nhất HPS, với cấu trúc một hoặc hai lõi xử lý ARM mang lại một tiện ích vô cùng thuận lợi trong các bài toán nhúng, đặc biệt phù hợp với các bài toán cần thời gian thực hoặc hệ điều hành. Trong phần nghiên cứu CNC-on-Chip của nhóm tác giả thì phần HPS đảm nhiệm vai trò giao tiếp máy lập trình và chức năng biên dịch trong bộ CNC-on-Chip của máy công cụ, phần thứ hai là FPGA (cho phép tái cấu hình hardware) đảm nhiệm chức năng nội suy, điều khiển vị trí các trục của máy công cụ. Họ SoC hiện nay tích hợp ARM Cortex A9, tốc độ hơn 800MHz như: Cyclone V, Arria V, Arria 10, Zynq và SmartFusion. Thành phần thứ hai FPGA có tài nguyên phong phú lên đến 110kLE cho phép người dùng cấu hình phần cứng, tùy biến các ứng dụng [1].

Kiến trúc giao tiếp giữa HPS và FPGA của các dòng SoC được nhà sản xuất thiết kế theo dạng cầu bắt tay [1]. Trên hình H.1 là dòng SoC của Altera. Trong đó, HPS-FPGA interfaces là cầu nối giữa hai thành phần trên.



H.1 Cấu trúc của SoC Altera [1]

Kể thừa dòng sản phẩm FPGA đã được khai thác sử dụng nhiều năm qua, nhiều IP cũng đã được cung cấp miễn phí từ nhà sản xuất, bên cạnh đó nhiều nhà thiết kế đã xây dựng IP trên nền FPGA cung cấp cho nhiều ứng dụng khác nhau. Truyền thông giữa FPGA với ARM [2] (hai thành phần này rời nhau) cũng được sử dụng ngay khi chưa có dòng sản phẩm SoC, chúng được ứng dụng cho các thiết kế CNC [3]. Nhờ tích hợp lõi Nios trên FPGA, một số tác giả đã thiết kế hệ thống giao tiếp dữ liệu giữa lõi xử lý mềm Nios với ngoại vi thông qua bus Avalon cho thấy hiệu suất trao đổi dữ liệu giữa chúng cũng rất hiệu quả [4]. Tuy

nhien sử dụng Nios trên FPGA làm tốn rất nhiều tài nguyên.

Các đề tài nghiên cứu khoa học liên quan về IP do Trung tâm Nghiên cứu & Đào tạo Thiết kế Vi mạch (ICDREC) thuộc khu Công nghệ Phần mềm ĐHQG triển khai từ 2006 – 2009. Cụ thể như: Thiết kế lõi IP giao tiếp bus I2C chế độ Master, Slaver, SDRAM, DSP, nghiên cứu thiết kế và thử nghiệm lõi IP biến đổi Fourier nhanh FFT [11] và nhiều nghiên cứu khác liên quan đến IP cho FPGA mà trong nước hầu như chưa đề cập đến vấn đề nghiên cứu IP cho dòng SoC.

Hầu hết các công trình nghiên cứu IP trong nước đã và đang thực hiện các nghiên cứu giữa FPGA và ngoại vi. Những IP này cũng được khá phổ biến hiện nay trong các ứng dụng của FPGA, nhất là lĩnh vực viễn thông [5], [2].

Cấu hình giao tiếp HPS và FPGA cũng đã được một số nhà thiết kế ngoài nước khai thác với những mục đích ứng dụng cụ thể. Dòng SoC Zynq 700 của Xilinx được sử dụng cho việc kiểm soát tốc độ truyền thông giữa hai thành phần trên, nhờ cầu AXI-HP cho phép trao đổi dữ liệu lên 1,2Gbps trên nền thời gian thực [6]. Một thiết kế IP truyền thông giao tiếp với các ngoại vi có yêu cầu tần số cao được thiết kế trên nền IP, nhằm khắc phục các nhược điểm lõi xử lý mềm bị hạn chế về tốc độ truyền thông [7]. IP PAC được thiết kế để theo dõi hiệu suất bus AXI nhằm tối ưu hóa dữ liệu truyền thông, hạn chế nút cổ chai trên SoC cũng được nghiên cứu và ứng dụng [8].

Lõi IP hầu hết được thiết kế dựa vào ngôn ngữ HDL. Trong thời gian gần đây, công cụ Qsys có nhiều ưu điểm cho việc thiết kế IP dựa thư viện sẵn có [9].

Trong bài viết này, nhóm tác giả nghiên cứu, thiết kế ứng dụng hệ thống giao tiếp giữa HPS và FPGA (IP_AXI_System) có lưu lượng truyền thông lên đến 64 bit, tốc độ xử lý 400MHz theo kiến trúc AXI tổng quát, nhằm đáp ứng cho trao đổi dữ liệu của bộ CNC-on-Chip mà nhóm tác giả đang nghiên cứu, qua đó phù hợp cho nhiều loại dữ liệu, kết nối nhiều ngoại vi, dựa trên ngôn ngữ Verilog và công cụ Qsys làm cầu nối trao đổi dữ liệu phù hợp cho hầu hết những bài toán nhúng sau này.

2. Nội dung chính

2.1 Kiến trúc giao tiếp giữa HPS và FPGA

Kiến trúc HPS-FPGA Interfaces là cầu nối giữa hai thành phần của một SoC, nó bao gồm ba cầu [1] HPS to- FPGA, FPGA- to- HPS và Lightweight HPS- to- FPAG chỉ ra trên hình H.2.

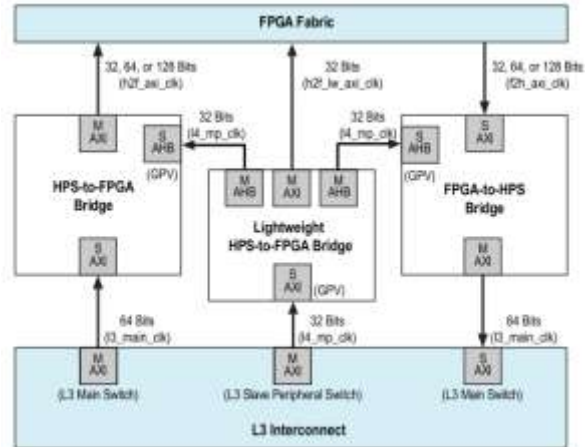
Kiến trúc này hỗ trợ các đường bus 32, 64, 128 bit thuận lợi cho việc thiết lập IP ứng dụng giao tiếp của hai thành phần trên SoC.

+ HPS-to-FPGA Bridge: Với thành phần master là phía HPS trên L3 Main Switch và phần slave bên FPGA cho phép master thực hiện cấu hình trong FPGA. Dữ liệu cho phép có thể 32, 64 hoặc 128 bit.

+ FPGA-to-HPS Bridge: Với thành phần master là phía FPGA và phần slave bên HPS cho phép master hiện các truy cập vào HPS trên ARM, hoặc các bộ

nhớ Cache. Dữ liệu cho phép có thể 32, 64 hoặc 128 bit.

+ Cầu lightweight HPS-to-FPGA cung cấp một giao diện hiệu suất thấp hơn cho FPGA. Giao diện này rất hữu ích cho việc truy cập các điều khiển và trạng thái thanh ghi của thiết bị ngoại vi mềm. Cầu cung cấp 2Mb không gian địa chỉ để truy cập vào phần logic, thiết bị ngoại vi và bộ nhớ thực thi trong FPGA.

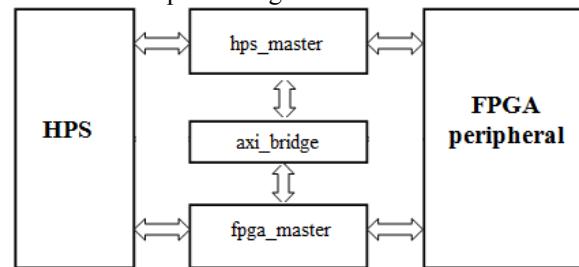


H.2 Kiến trúc của giao tiếp HPS-FPGA trong SoC [1]

Cầu lightweight HPS-to-FPGA có ba giao diện tổng thể và tích hợp hệ thống. Các giao diện tổng thể kết nối với FPGA cung cấp một giao diện Lightweight từ HPS logic tùy chỉnh trong FPGA. Hai giao diện tổng thể khác là kết nối với cầu HPS- to-FPGA và FPGA-to-HPS, cho phép truy cập vào thanh ghi GPV (Global Programmers View) ở mỗi cầu.

2.2 Đề xuất cấu trúc IP_AXI_System

Để đáp ứng nhu cầu giao tiếp của chức năng biên dịch và chức năng nội suy của bộ CNC-on-Chip cũng như mọi bài toán truyền thông giữa HPS và FPGA thì yêu cầu kiến trúc IP_AXI_System phải đáp ứng được truyền thông toàn bộ các thành phần trong HPS cũng như các thành phần trong FPGA.



H.3 Đề xuất kiến trúc IP_AXI_System trong SoC

Đồng thời, giải quyết nhiều vấn đề về tắc nghẽn đường truyền nội bộ giữa hai thành phần này, các hiện tượng nút cổ chai cũng được kiến trúc IP_AXI_System tối ưu hóa trong quá trình truyền thông. Qua đó, đề xuất cấu trúc của nhóm tác giả như trên hình H.3 bao gồm ba thành phần trao đổi dữ liệu như sau:

- Thành phần hps_master: Cho phép chuyển đổi dữ liệu tối đa 128 bit giữa HPS và FPGA trong đó vai trò master là phần HPS. Đề xuất ở đây sử dụng JTAG to Avalon Master Bridge nằm trong nhóm cầu ánh xạ bộ nhớ của công cụ Qsys để làm thành phần này.

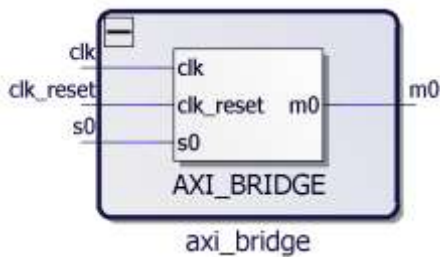
- Thành phần fpga_master: Tương tự như thành phần hps_master, thành phần này cũng được đề xuất sử

dùng JTAG to Avalon Master Bridge trong nhóm cầu ảnh xạ bộ nhớ của công cụ Qsys.

- Thành phần axi_bridge: Có chức năng quản lý hai thành phần trên và cho phép trao đổi dữ liệu giữa HPS và FPGA. Đề xuất ở đây sử dụng thư viện Axi bridge trong nhóm cầu ảnh xạ bộ nhớ trong Qsys để đảm nhiệm vai trò chức năng giao tiếp và điều khiển hệ thống.

2.3 Thiết kế các thành phần của IP_AXI_System

- Thành phần axi_bridge: Cấu hình giao tiếp với bộ nhớ sử dụng axi bridge với tín hiệu vào lưu lượng truyền thông 64 bit đáp ứng dữ liệu sau khi biên dịch, địa chỉ rộng 11 bit. Giao tiếp slave và master có ID 8bit, các điều khiển đọc viết 16 bit. Chương trình được xây dựng file axi_bridge.v dựa trên chuẩn "altera axi bridge". Kết quả thiết kế thành phần này thể hiện sơ đồ khối trên hình H.4.

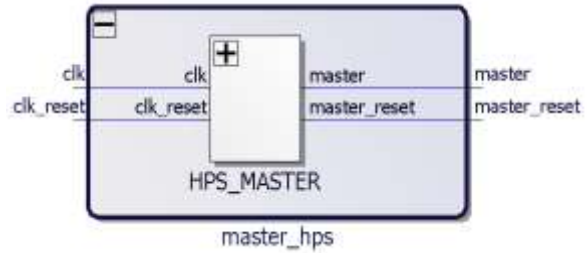


H.4 Kiến trúc axi_bridge của IP_AXI_System

Các thông số thiết kế thành phần axi_bridge bao gồm tham số cấu hình thể hệ AXI4, xác lập các ID cho master và slave và nhiều tham số khác trong tập tin axi_bridge.v. Chức năng của axi_bridge điều phối truyền thông dữ liệu hai chiều giữa phần mềm trên HPS và phần cứng trên FPGA.

- Thành phần master_hps: Nhằm điều khiển các giao tiếp ngoại vi, ở đó HPS đóng vai trò là master, các thành phần trên FPGA đóng vai trò các slave. Trong đó địa chỉ 32 bit, dữ liệu đọc, viết 64 bit và các bit điều khiển. Chương trình được thiết kế trong tập tin

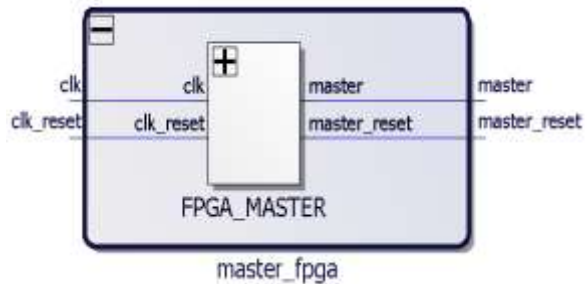
master_hps.v dựa trên chuẩn "alter jtag to avalon master". Kết quả sau khi thiết kế thể hiện trên sơ đồ khối trên hình H.5.



H.5 Kiến trúc master_hps của IP_AXI_System

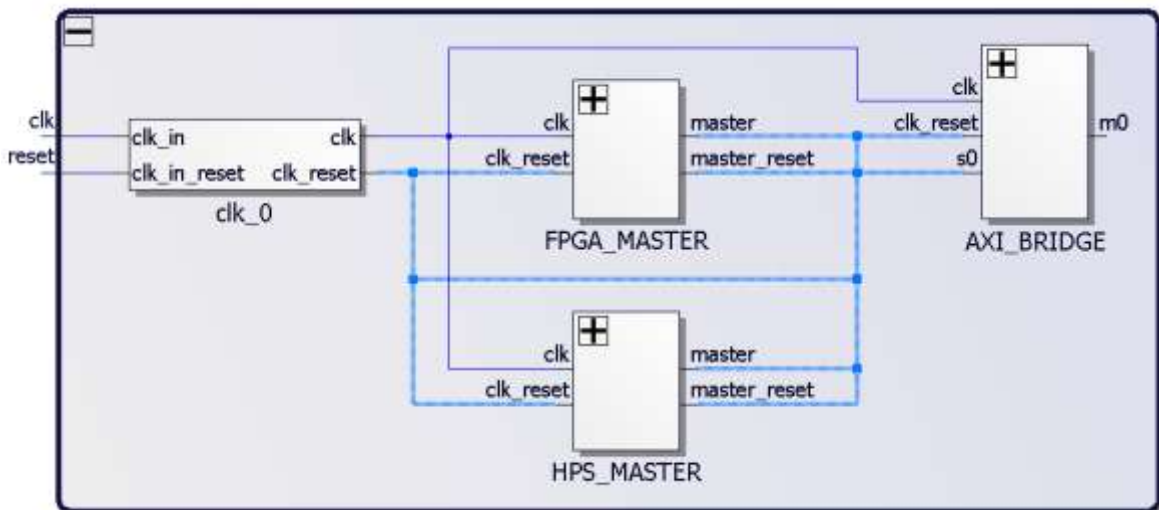
Thành phần master_hps đảm nhiệm chức năng giao tiếp hệ thống vai trò master ở HPS, các lệnh điều khiển ra lệnh từ HPS tới FPGA.

- Thành phần master_fpga: Dùng để điều khiển giao tiếp với các thành phần ngoại vi trên FPGA. Trong đó địa chỉ 32 bit, dữ liệu đọc/viết 64 bit và các bit điều khiển. Chương trình được thiết kế trong tập tin master_fpga.v dựa trên chuẩn IP "alter jtag to avalon master". Kết quả sau khi thiết kế thể hiện trên sơ đồ khối trên hình H.6.



H.6 Kiến trúc master_FPGA của IP_AXI_System

Thành phần master_fpga đảm nhiệm chức năng phân phối các ngoại vi trên FPGA, thực thi các ngắt từ ngoại vi FPGA.



H.7 Cấu trúc IP_AXI_System

Tổng hợp từ các thành phần đã thiết kế sử dụng công cụ Qsys để tích hợp các thành phần trên vào một dự án cùng với nguồn xung clock. Hệ thống truyền thông giữa HPS và FPGA (IP_AXI_System) được thiết kế

như trên hình H.7. Cho thấy khả năng trao đổi dữ liệu phù hợp với bài toán trao đổi dữ liệu cho bộ CNC-on-Chip mà nhóm tác giả đang nghiên cứu. Không chỉ phục vụ cho bộ CNC-on-Chip mà IP_AXI_System sẽ

được tích hợp trong thư viện Qsys nhằm đáp ứng những các bài toán nhúng trên SoC sau này.

2.4 Cấu hình IP_AXI_System trên công cụ Qsys

Nhiều tác giả, đã khai thác khả năng cấu hình phần cứng và phần mềm, trong đó lõi Nios được ứng dụng khá nhiều [5]. Khi đó, hệ thống giao tiếp Nios và ngoại vi FPGA cũng trở nên đơn giản bởi thành phần lõi Nios được thiết kế ngay trên FPGA. So với Nios thì, HPS có kiến trúc lõi cứng ARM được tích hợp trên SoC nên dữ liệu trao đổi rất lớn phù hợp với nhiều bài toán nhúng phức tạp, đặc biệt là CNC-on-Chip. Vì thế, việc thiết lập cấu giao tiếp giữa chúng là cần thiết.

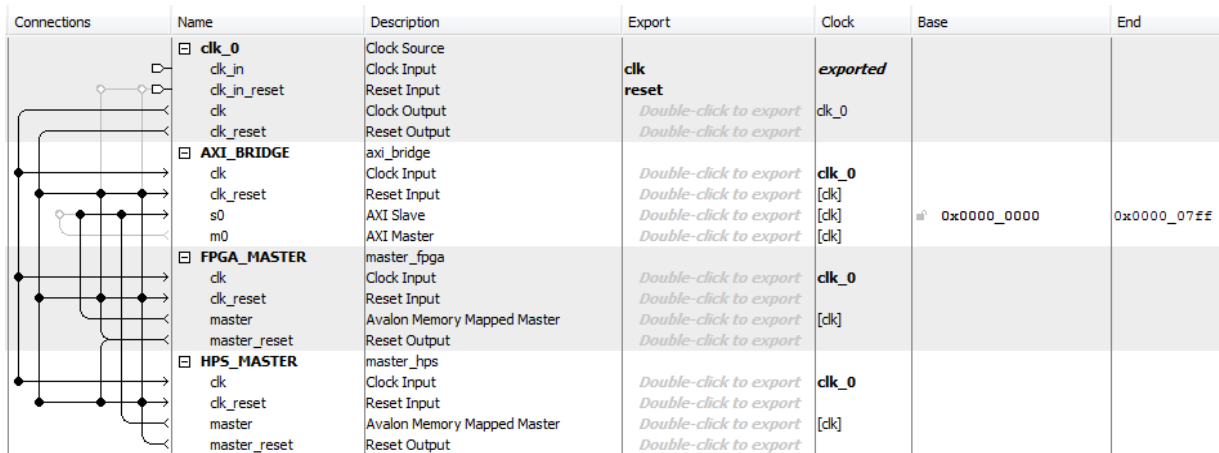
Sau đây, nhóm tác giả trình bày thiết kế hệ thống truyền thông giữa HPS và FPGA gọi tắt là IP_AXI_System bằng công cụ Qsys cùng với các thành phần đã tạo ra ở trên. Từ menu Tools của Quartus lựa chọn Qsys để tạo ra cửa sổ thực hiện cấu hình giao tiếp như trên hình H.8. Trong cửa sổ Qsys thực hiện chọn các thành phần AXI_BRIDGE, FPGA_MASTER, HPS_MASTER đã thiết kế từ phần

trên. Chèn các thành phần này và tùy chỉnh tham số khi cần thiết.

Tất cả các thành phần được thiết kế trong FPGA đều được kết nối với nguồn xung clk_0. Các đối tượng ở đây đều là phần bên trong SoC nên trên cột Export chỉ có clk và reset được kết nối ngoại vi ra bên ngoài.

Đối với các đường bus dữ liệu, sử dụng các kết nối giữa master và slave giữa của các thiết kế. Dĩ nhiên, khi có thêm HPS và điều khiển trên FPGA việc kết nối này là "tùy biến". Trên hệ thống Qsys hình H.8 nhóm tác giả sử dụng các cấu hình kết nối để tạo ra hệ thống trao đổi dữ liệu giữa các khối chức năng và nó sẽ được cấu hình lại cho phù hợp khi có HPS.

Một số kết nối trên đối tượng là không cần thiết khi người thiết kế không sử dụng thì mặc định sẽ không được kết nối đến bus IP_AXI_System. Vì bài viết này chỉ đề cập đến việc xây dựng hệ thống giao tiếp giữa HPS và FPGA nên ngoại vi cũng như thành phần HPS không thể hiện trên hình H.8 mà sẽ được trình bày trên hình H.11 ở phần sau.



Connections	Name	Description	Export	Clock	Base	End
clk_0	clk_in	Clock Source	clk	exported		
	clk_in_reset	Clock Input	reset			
	clk	Reset Input	Double-click to export	clk_0		
	clk_reset	Clock Output	Double-click to export			
AXI_BRIDGE	axi_bridge	Reset Output	Double-click to export			
	clk	AXI Slave	Double-click to export	clk_0		
	clk_reset	AXI Master	Double-click to export	[dk]		
	s0	AXI Slave	Double-click to export	[dk]	0x0000_0000	0x0000_07FF
FPGA_MASTER	m0	AXI Master	Double-click to export	[dk]		
	master_fpga	AXI Slave	Double-click to export			
	clk	AXI Master	Double-click to export	clk_0		
	clk_reset	AXI Slave	Double-click to export	[dk]		
HPS_MASTER	master	Avalon Memory Mapped Master	Double-click to export			
	master_reset	Reset Output	Double-click to export			
	master_hps	AXI Slave	Double-click to export	clk_0		
	clk	AXI Master	Double-click to export	[dk]		
HPS_MASTER	clk	Avalon Memory Mapped Master	Double-click to export			
	clk_reset	Reset Input	Double-click to export			
	master	Avalon Memory Mapped Master	Double-click to export			
	master_reset	Reset Output	Double-click to export			

H.8 Cấu hình IP_AXI_System bằng công cụ Qsys

Kết quả cấu hình sẽ được lưu trữ trong thư viện IP và phục vụ cho các bài toán điều khiển nhúng nói chung, nhằm giao tiếp giữa hai thành phần HPS và FPGA. Đây là cấu hình mở với đầy đủ ba thành phần của hệ thống cầu được mô tả tên hình H.3 mục 2.2 mà nhóm tác giả đã đề xuất. Đồng thời phù hợp với kiến trúc AXI của SoC mà nhà thiết kế đã kiến trúc sẵn phần cứng trên SoC.

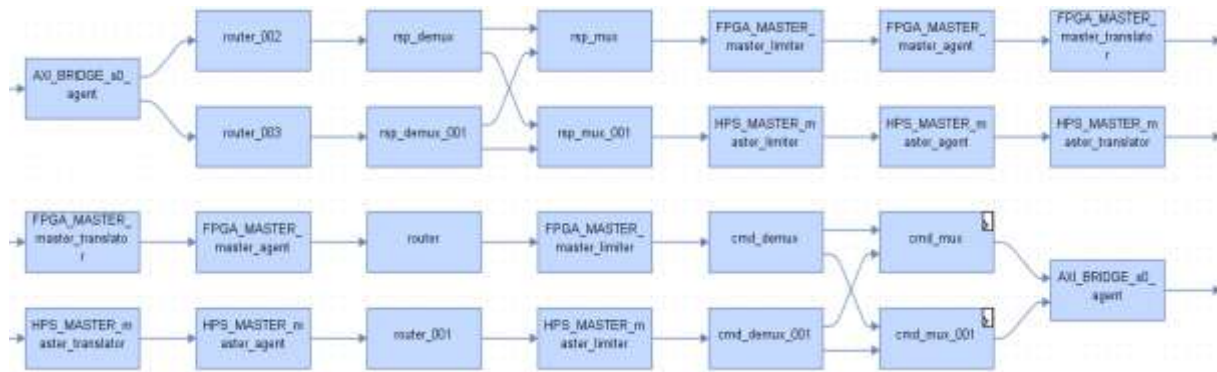
Kết quả IP_AXI_System được sử dụng cho truyền thông nội bộ của bộ CNC-on-Chip. Trong một số ứng dụng khác người dùng có thể sử dụng một hoặc hai trong ba thành phần trên nhằm tiết kiệm tài nguyên trên SoC khi cấu hình mà vẫn đáp ứng được truyền thông dữ liệu giữa HPS và FPGA.

Một IP_AXI_System hoàn thiện là một hệ thống có đủ cả ba thành phần đã đề xuất trên và được nhóm tác giả thiết kế từng thành phần trước đó. Trên hình H.9,

thể hiện kết quả điều khiển truyền thông hai chiều của IP_AXI_System, khi chạy cấu hình bằng công cụ Qsys của Quartus.

Một hệ thống sau khi xây dựng hoàn thiện như trên hình H.8, nhóm tác giả tiến hành chạy "map" RTL bằng Quartus 14.0 cho kết quả như trên hình H.9. Từ kết quả này cho thấy bảo đảm được luồng dữ liệu hai chiều đồng thời, được điều khiển bởi các master của từng thành phần và đáp ứng trao đổi dữ liệu hai chức năng biên dịch và nội suy của bộ CNC-on-Chip và mở rộng cho các bài toán nhúng trên SoC.

Cầu AXI_BRIDGE đều phối quá trình hoạt động của FPGA_MASTER và HPS_MASTER theo cả hai hướng dữ liệu truyền tải trong trường hợp này là 64 bit. Để thực hiện kiểm chứng kết quả hoạt động, IP_AXI_System sẽ được thiết kế cùng với lõi HPS và các điều khiển trên FPGA trong phần sau.

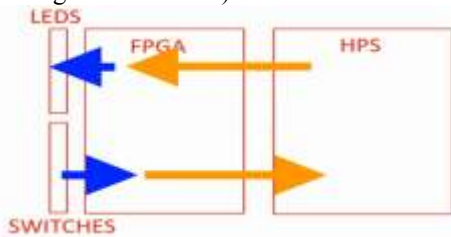


H.9 Sơ đồ kết nối IP_AXI_System theo 2 luồng điều khiển

2.5 Đánh giá kết quả

Trong phần này, nhóm tác giả thử nghiệm kết quả thiết kế IP_AXI_System trên KIT DE1-SoC bằng một ứng dụng đơn giản giao tiếp 32 bit (Kit chỉ hỗ trợ GPIO 32 bit) dữ liệu nhận từ 10 Switch để làm sáng 10 LED. Cấu trúc kiểm tra của hệ thống được trình bày như trên hình H.10, sử dụng bộ nhớ Ram on Chip trên SoC và SDRam có sẵn trên KIT làm nơi lưu trữ dữ liệu của ứng dụng.

Thiết kế hệ thống kiểm tra IP_AXI_System được xây dựng trên phần mềm Quartus (thiết lập cấu hình hardware) và phần mềm ARM DS-5 (xây dựng chương trình software)



H.10 Sơ đồ thể hiện hệ thống điều khiển nhúng SoC - Thiết kế hardware cho hệ thống thực hiện bằng công cụ Qsys của Quartus như sau:

+ Chèn lõi xử lý Hard Processor Systems,

- + IP_AXI_System đã thiết kế,
- + Memory on Chip
- + System ID peripheral
- + IO (Switch và LED)

Trong đó, nguồn clock và reset được kết nối đầy đủ tới các thành phần trong hệ thống, axi_master của HPS với các thành phần axi_slave của FPGA là Switch và Led. Đây chỉ là ví dụ đơn giản để kiểm chứng khả năng làm việc của IP_AXI_System trong những ứng dụng lớn hơn thì việc kết nối cũng sẽ phức tạp hơn nhiều.

Đặt các xuất nhập ngoại vi cho SW và LED trên cột Export tương ứng với địa chỉ trên KIT DE1-SoC và định không gian địa chỉ trên cột Base và End theo sản phẩm SoC dòng 5CSEMA5F31C có trên KIT.

Ngoài ra ở đây sử dụng hps_io và memory là bộ nhớ ngoài để lưu trữ trao đổi dữ liệu được tích hợp sẵn trên KIT 1Gb SDRAM. Kết quả thực hiện cấu hình hardware trên công cụ Qsys kiểm tra khả năng làm việc của IP_AXI_System được thiết lập với các IP ngoại vi FPGA như trên hình H.11. Trong đó, thành phần HPS là hps_0 và thành phần FPGA là SWITCH, LED.

Connections	Name	Description	Export	Clock	Base	End	IRQ
	clk_0	Clock source		exported			
	hps_0	Altera V/Cydone V Hard Processor System					
	hps_0_f2h_stm_hw_events	Conduit	hps_0_f2h_stm_hw_ev...				
	memory	Conduit	memory				
	hps_io	Conduit	hps_io				
	h2f_reset	Reset Output	hps_0_h2f_reset				
	h2f_axi_clock	Clock Input	hps_0_h2f_axi_clock	clk_0			
	h2f_axi_master	AXI Master	hps_0_h2f_axi_master	[h2f_axi_clock]			
	h2f_axi_slave	AXI Slave	hps_0_h2f_axi_slave	clk_0			
	h2f_jw_axi_clock	Clock Input	hps_0_h2f_jw_axi_clock	clk_0			
	h2f_jw_axi_master	AXI Master	hps_0_h2f_jw_axi_master	[h2f_jw_axi_clock]			
	f2h_irq0	Interrupt Receiver	hps_0_f2h_irq0				IRQ 0
	f2h_irq1	Interrupt Receiver	hps_0_f2h_irq1				IRQ 31
	AXI_BRIDGE	AXI Bridge					
	ck	Clock Input	hps_0_axi_bridge_ck	clk_0			
	ck_reset	Reset Input	hps_0_axi_bridge_ck_reset	[ck]			
	sl	AXI Slave	hps_0_axi_bridge_sl	[ck]			
	ml	AXI Master	hps_0_axi_bridge_ml	[ck]			
	HPS_MASTER	JTAG to Avalon Master Bridge					
	ck	Clock Input	hps_0_hps_master_ck	clk_0			
	ck_reset	Reset Input	hps_0_hps_master_ck_reset	[ck]			
	master	Avalon Memory Mapped Master	hps_0_hps_master_master	[ck]			
	master_reset	Reset Output	hps_0_hps_master_master_reset				
	FPGA_MASTER	JTAG to Avalon Master Bridge					
	ck	Clock Input	hps_0_fpga_master_ck	clk_0			
	ck_reset	Reset Input	hps_0_fpga_master_ck_reset	[ck]			
	master	Avalon Memory Mapped Master	hps_0_fpga_master_master	[ck]			
	master_reset	Reset Output	hps_0_fpga_master_master_reset				
	SWITCH	PIO (Parallel I/O)		clk_0	0x0001_0010	0x0001_001f	
	LED	PIO (Parallel I/O)		clk_0	0x0000_0000	0x0000_000f	
	sysid_qsys	System ID Peripheral		clk_0	0x0001_0000	0x0001_000f	
	onchip_memory2_0	On-Chip Memory (RAM or ROM)		clk_0	0x0000_0000	0x0000_0fff	

H.11 Cấu hình hardware IP_AXI_System bằng công cụ Qsys

Sau khi kết nối các đối tượng của hệ thống Qsys ta tiến hành định địa chỉ không gian nhớ. Việc định địa chỉ này được gọi là ánh xạ bộ nhớ khi hệ thống làm việc. Các địa chỉ không gian nhớ theo quy ước của nhà sản xuất tương ứng với từng thành phần. Nhóm

tác giả định địa chỉ cho hệ thống thiết kế này như trên bảng B.1, việc định địa chỉ này được sử dụng cho chương trình nhúng sau này. Trong đó "N" không sử dụng địa chỉ do thành phần AXI_BRIDGE không sử dụng trong trường hợp này.

B.1 Bảng phân chia địa chỉ giao tiếp HPS và FPGA

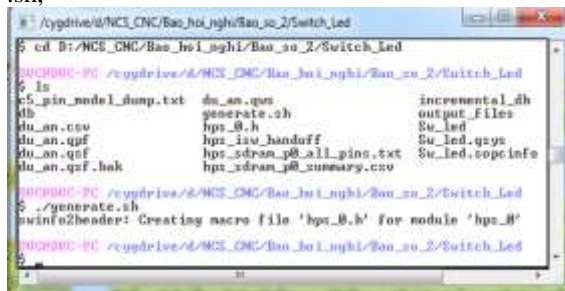
Luồng dữ liệu	hps_0.f2f_axi_master	hps_0.f2f_lw_axi_master	HPS_MASTER.master	FPGA_MASTER.master	AXI_BRIDGE.m0
onchip_memory_2_0.s1	0x0000_0000 - 0x0000_0fff				N
sysid_qsys.control_slave		0x0001_0000 - 0x0001_0007		0x0001_0000 - 0x0001_0007	N
SWITCH.s1		0x0001_0010 - 0x0001_001f		0x0001_0010 - 0x0001_001f	N
LED.s1		0x0000_0000 - 0x0000_000f	0x0000_0000 - 0x0000_000f		N
hps_OBRIDGE.s0	N	N	N	N	N

- Thực hiện cấu hình software cho thành phần HPS để kiểm tra chương trình ứng dụng. Trước hết nhóm tác giả xây dựng tập tin **generate.sh** trong cùng thư mục với các tập tin khác như .sopcinfo, .c, .v...

```
#!/bin/sh
sopc-create-header-files \
"/Sw_led.sopcinfo" \
--single hps_0.h \
--module hps_0
```

Chức năng của tập tin generate.sh dùng tạo ra tập tin head có tên là hps_0.h cùng đường dẫn thư mục bằng lệnh **./generate.sh** ngay trong cửa sổ Embedded Command Shell. Tập tin heard này chứa toàn bộ thông tin về IP_AXI_System mà nhóm tác giả đã thiết kế trên Qsys, các thành phần Sw và Led trên FPGA được thực hiện liên kết với HPS.

Một hệ thống nhúng cần có tập tin heard, nó cần cho việc định nghĩa địa chỉ của thành phần HPS và FPGA, các địa chỉ này sẽ được cung cấp cho chương trình C. Kết quả xây dựng tập tin head trên hình H.12 có tên **hps_0.h** được tạo ra cùng thư mục với các tập tin .c, .sh,

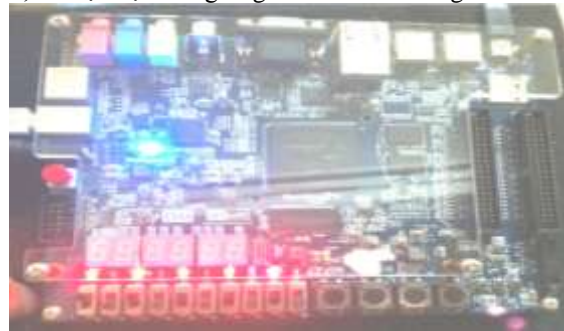


H.12 Tạo file header hps_0.h

Nhóm tác giả xây dựng chương trình phần mềm ứng dụng bằng ngôn ngữ C (ở đây đặt tên **main.c**) để kiểm tra hoạt động của IP_AXI_System bật tắt SW nào tương ứng LED đó sáng.

Dữ liệu truyền thông ở đây là 32 bit theo thiết kế cấu hình ban đầu. Tuy nhiên với KIT DE1-SoC chỉ tích hợp 10 LED và 10 SW nên việc biểu diễn trên Kit cho kết quả tương ứng cho 10 bit như hình H.13 cho thấy

kết quả hoạt động đúng như chương trình ứng dụng C đã viết trên bằng cách lấy nội dung của địa chỉ SW gán cho các LED. Trên hình H.13 là các SW 1, 3, 5, 7, 9 được bật tương ứng với các LED sáng.



H.13 Tạo file thực thi nhị phân và chạy trên KIT

Qua thực nghiệm cho thấy khẳng định thiết kế IP_AXI_System hoạt động được và hoàn toàn thực hiện được nhiệm vụ trao đổi dữ liệu giữa hai chức năng biên dịch và nội suy của bộ CNC-on-Chip, đồng thời đáp ứng các bài toán nhúng trên SoC.

3. Kết luận

Thiết kế IP_AXI_System giữa thành phần HPS và FPGA trong SoC là điều cần thiết trong các thiết kế hệ thống nhúng. Việc trao đổi dữ liệu giữa HPS và FPGA tùy thuộc vào hệ thống thiết kế giao tiếp này và từng bài toán ứng dụng cụ thể. IP_AXI_System mà nhóm tác giả thiết kế cho phép dữ liệu trao đổi tối đa 64 bit, thực hiện hai luồng và hai chiều trao đổi dữ liệu, phù hợp với mọi thành phần điều khiển trong SoC.

Bài viết này đã thực hiện được các đóng góp so với trước đây nhiều tác giả chỉ thiết kế các giao tiếp điều khiển trên FPGA với ngoại vi như SRAM, I2C, I/O, Nios [4], [7]...

- Thiết kế được xây dựng từ cấu hình thành phần dạng modul tạo nên hệ thống giao tiếp giữa HPS và FPGA (IP_AXI_System) lưu trữ thư viện phục vụ cho các ứng dụng khác.

- Thiết kế được hệ thống IP_AXI_System phục vụ cho bài toán CNC-on-Chip.
- Bước đầu đưa dòng sản phẩm SoC vào ứng dụng tại Việt Nam nhất là lĩnh vực điều khiển nhằm mang lại lợi ích về tốc độ, chính xác và khả năng tích hợp các chương trình ứng dụng lớn trên một chip.

Tài liệu tham khảo

- [1] Altera, *Cyclone V Device Handbook Volume 3 : Hard Processor*, System Technical Reference Manual, vol. 3. 2014.
- [2] J. Fei, R. Deng, Z. Zhang, and M. Zhou, *Research on Embedded CNC Device Based on ARM and FPGA*, Procedia Eng., vol. 16, pp. 818–824, Jan. 2011.
- [3] T. Wang, L. Wang, and Q. Liu, *A three-ply reconfigurable CNC system based on FPGA and field-bus*, Int. J. Adv. Manuf. Technol., vol. 57, no. 5–8, pp. 671–682, Apr. 2011.
- [4] A. Aysu and P. Schaumont, *Hardware/software co-design of physical unclonable function based authentications on FPGAs*, Microprocess. Microsyst., 2015.
- [5] V. Nagarajan, R. Hariharan, V. Srinivasan, R. S. Kannan, P. Thinakaran, V. Sankaran, B. Vasudevan, R. Mukundrajana, N. C. Nachiappan, A. Sridharan, K. P. Saravanan, V. Adhinarayanan, and V. V. Sankaranarayanan, *SCOC IP Cores for Custom Built Supercomputing Nodes*, 2012 IEEE Comput. Soc. Annu. Symp. VLSI, pp. 255–260, Aug. 2012.
- [6] Q. Tang, L. Jiang, X. X. Liu, and Q. Dai, *A real-time updatable FPGA-based architecture for fast regular expression matching*, Procedia Comput. Sci., vol. 31, no. Itqm, pp. 852–859, 2014.
- [7] S. Chebiyyam, R. Maddasani, S. R. Gaddala, and S. Y. Pasha, *Design of High Speed Serial Communication Protocol for FPGA based SoC Applications*, vol. 3, no. 2, pp. 1–4, 2013.
- [8] H. M. Kyung, G. H. Park, J. Wook Kwak, T. J. Kim, and S. B. Park, *Design and implementation of Performance Analysis Unit (PAU) for AXI-based multi-core System on Chip (SOC)*, Microprocess. Microsyst., vol. 34, no. 2–4, pp. 102–116, 2010.
- [9] A. Corporation, *Qsys System Design Components*, 2014.
- [10] <http://news.vnu.edu.vn>



Đỗ Văn Cận, sinh năm 1981, nhận bằng thạc sỹ về *Tự động hóa* của Đại học Đà Nẵng năm 2009. Từ năm 2004 đến nay là giảng viên của Khoa Kỹ thuật và Công nghệ, trường Đại học Quy Nhơn. Hiện nay, đang NCS Đại học Đà Nẵng. Công trình

được nghiên cứu tại viện Kỹ thuật điều khiển và Tự động hóa, Trường Đại học Bách khoa Hà Nội.



TSKH. Nguyễn Phùng Quang sinh 1953, nhận học vị Dipl.-Ing. (Uni.) tháng 9/1975, Dr.-Ing. tháng 11/1991 và Dr.-Ing. habil. tháng 4/1994, tất cả đều tại TU Dresden (TUD, CHLB Đức). Có nhiều năm làm việc trong thực tiễn công nghiệp Đức, góp phần cho ra đời các biến tần REFU 402 Vectovar, RD500 (công ty REFU

Elektronik), Simovert 6SE42, Master Drive MC (tập đoàn Siemens).

1996-1998 là giảng viên của TUD, tại đây tháng 10/1997 được công nhận là Privatdozent. Về nước đầu 1999 và là giảng viên của ĐHBK Hà Nội từ đó đến nay. Tháng 2/2004 được TUD phong tặng chức danh Honorarprofessor, 11/2004 nhận chức danh Phó Giáo sư và 11/2009 Giáo sư về Tự động hóa của ĐHBK Hà Nội.

Là tác giả / đồng tác giả của hơn 150 bài báo, báo cáo hội nghị trong và ngoài nước. Là tác giả / đồng tác giả của 7 đầu sách chuyên khảo và tham khảo, trong đó có 3 quyển bằng tiếng Đức và 1 quyển tiếng Anh “*Vector Control of Three-Phase AC Machines – System Development in the Practice*” do nhà xuất bản Springer in năm 2008, tái bản lần 1 tháng 6/2015.

Các lĩnh vực nghiên cứu: điều khiển truyền động điện, điều khiển chuyển động và robot, điều khiển vector cho các loại máy điện, điều khiển điện tử công suất, điều khiển các hệ thống năng lượng tái tạo (sức gió, mặt trời), hệ thống điều khiển số, mô hình hóa và mô phỏng.



Đoàn Quang Vinh sinh năm 1962, tốt nghiệp Kỹ sư năm 1986 tại trường đại học Cơ – Điện Pilsner, Tiệp Khắc và nhận bằng tiến sĩ năm 1996 tại Đại học West Bohemia, cộng hòa Séc. Hiện nay, công tác tại Đại học Đà Nẵng. Lĩnh vực nghiên cứu: Truyền động điện, điện tử công suất, mô hình hóa hệ thống điều khiển...