BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT

THÀNH PHỐ HỒ CHÍ MINH

HCMUTE

CÔNG TRÌNH NGHIÊN CỨU KHOA HỌC CẤP TRƯỜNG

NGHIÊN CỨU GIAO TIẾP EPGA VỚI MODULE SIM VÀ VIẾT CHƯƠNG TRÌNH ĐIỀU KHIỂN THIẾT BỊ TỪ XA

MÃ SỐ: T2014 - 07

SKC005640

Tp. Hồ Chí Minh, 2015

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT
THÀNH PHỐ HỒ CHÍ MINH

BÁO CÁO TỔNG KẾT

ĐỀ TÀI KH&CN CẤP TRƯỜNG

NGHIÊN CỨU GIAO TIẾP FPGA VỚI MODULE SIM VÀ VIẾT CHƯƠNG TRÌNH ĐIỀU KHIỂN THIẾT BỊ TỪ XA

MÃ SỐ: T2014 - 07

Chủ nhiệm đề tài: GV.THS. TRƯƠNG THỊ BÍCH NGÀ

TP. HCM, 03/2015

DANH SÁCH

NHỮNG THÀNH VIÊN THAM GIA NGHIÊN CỨU ĐỀ TÀI

VÀ ĐƠN VỊ PHỐI HỢP CHÍNH

1. Các thành viên tham gia gồm có

GVC.ThS. Nguyễn Đình Phú

2. Các đơn vị phối hợp

MỤC LỤC

Mở đầu . 11
Tình hình nghiên cứu trong và ngoài nước..... 11
Tính cấp thiết – mục tiêu. 11
Cách tiếp cận. 11
Phương pháp nghiên cứu. 12
Đối tượng và phạm vi nghiên cứu 12
Nội dung nghiên cứu 12
Chương 1: CÔNG NGHỆ FPGA
1.1 CÔNG NGHỆ FPGA 13
1.1.1 Cấu trúc của FPGA 13
1.1.2 Khối logic có thể định cấu hình..... 14
1.1.3 Các module 14
1.1.4 Các lõi của FPGA 14
1.2 FPGA CỦA ALTERA 15
1.2.1 Khối mảng logic (LAB –Logic Array Block) 15
1.2.2 Module Logic thích nghi ALM.....16
1.2.3 Các chức năng tích hợp 18
1.3 FPGA CỦA XILINX.....18
1.3.1 Các khối Logic có thể định cấu hình CLB 18
1.3.2 Chuỗi liên tiếp SOP.....20
1.3.3 Cấu trúc FPGA truyền thông và cấu trúc ASMBL.....22
Chương 2: GIỚI THIỆU KIT DILIGENT NEXYS 2 SPARTAN-3E 25
2.1 TỔNG QUAN VỀ BOARD NEXYS 2 25
2.2 NGUỒN CUNG CẤP CHO BOARD 26
2.3 CẤU HÌNH FPGA VÀ PLATFORM FLASH 27
2.4 CLOCK.....28
2.5 CỔNG NỐI TIẾP 31
2.6 BỘ NHỚ 31
Chương 3: THIẾT KẾ PHẦN CỨNG 33

3.1 SƠ ĐỒ KHỐI 33
3.2 CHI TIẾT TỪNG KHỐI NGOẠI VI 33
3.2.1 MODULE SIM 900 33
3.2.2 KHỐI CÔNG SUẤT 37
3.2.3 KHỐI HIỂN THỊ LCD 38
CHƯƠNG 4: THIẾT LẬP CẤU HÌNH CHO FPGA VÀ LƯU ĐỒ CHƯƠNG TRÌNH ĐIỀU KHIỂN 41
4.1 GIỚI THIỆU PHẦN MỀM EDK CỦA HÃNG XILINX 41
4.2 THIẾT KẾ CẤU HÌNH TRÊN FPGA DÙNG XPS 41
4.2.1 Yêu cầu 41
4.2.2 Sơ đồ phần cứng yêu cầu 41
4.2.3 Các bước thực hiện 42
4.3 THIẾT KẾ HỆ THỐNG NHÚNG TRÊN FPGA DÙNG SDK 57
4.3.1 Xây dựng các yêu cầu điều khiển theo tin nhắn 57
4.3.2 Lưu đồ chương trình 57
4.3.3 Xây dựng phần mềm 58
KẾT LUẬN 65
KIẾN NGHỊ VÀ HƯỚNG PHÁT TRIỂN 65
TÀI LIỆU THAM KHẢO 66

Danh mục hình

Hình 1-1: Cấu trúc FPGA 13
Hình 1-2: Các khối CLB 14
Hình 1-3: Sơ đồ khối của một module logic 14
Hình 1-4: Khái niệm chức năng lõi phần cứng trong FPGA 15
Hình 1-5: Sơ đồ khối của cấu trúc LAB của Stratix II và ALM 16
Hình 1-6: Sơ đồ khối ALM của Stratix II 17
Hình 1-7: Các cấu hình có thể có của LUT trong ALM của kiểu bình thường 17
Hình 1-8: Mở rộng ALM để tạo ra hàm SOP 7 biến trong kiểu LUT mở rộng 18
Hình 1-9: Sơ đồ khối của FPGA Stratix II 19
Hình 1-10: Minh họa các cấp logic định cấu hình từ tế bào logic cho đến CLB 20
Hình 1-11: Cách dung chuỗi nối tiếp để mở rộng biểu thức CLB 22
Hình 1-12: Tích hợp nhiều chức năng IP 23
Hình 1-13: Minh họa cấu trúc ASMBL của FPGA platform 24

Hình 2-1: Sơ đồ khối và đặc điểm của Nexys 2 25
Hình 2-2: Sơ đồ khối nguồn cung cấp Nexys 2 26
Hình 2-3: Mạch lập trình Nexys 2 27
Hình 2-4: Mạch lập trình Board Nexys 2 28
Hình 2-5: Thiết bị I /O Board Nexys 2 29
Hình 2-6: Mạch và thiết bị I / O của Board Nexys 2 29
Hình 2-7: sơ đồ nguyên lý của các cổng I/O mở rộng của kit 30
Hình 2-8 : cổng I/O thực tế 30
Hình 2-9: Mạch cổng nối tiếp Nexys 2 31
Hình 2-10: Mạch bộ nhớ Nexys 2 32

Hình 3-1: Sơ đồ khối của hệ thống 33
Hình 3-2: Module sim 900 34
Hình 3-3: Sơ đồ chân của module sim 900 36
Hình 3-4: Mạch nguồn cung cấp cho module sim 900 37
Hình 3-6: Mạch nguyên lý của mạch công suất 38
Hình 3-7: Sơ đồ mạch in của mạch công suất 38
Hình 3-8: Module LCD 16x2 39
Hình 3-9: Module PmodCLB 39

Hình 4-1: Sơ đồ phần cứng 42
Hình 4-2: Xóa LED_7SEGMENT 43
Hình 4-3: Chọn chế độ xóa IP 44
Hình 4-4: Cài đặt cấu hình IP cho RS_232_PORT_PC 44
Hình 4-5: Sau khi đổi tên thành RS_232_PORT_PC 44
Hình 4-6: Thêm IP UART 45
Hình 4-7: Cài đặt cấu hình cho RS232_PORT 45
Hình 4-8: Thêm IP OUT0 45
Hình 4-9: Cài đặt cấu hình cho OUT0 46

Hình 4-10: Thêm IP timer delay 46
Hình 4-11: Cài đặt cấu hình cho IP timer delay 47
Hình 4-12: Thêm IP ngắt xps_intc_0 47
Hình 4-13: Cài đặt cấu hình IP ngắt xps_intc_0 47
Hình 4-14: Chọn chế độ kết nối SPLB 48
Hình 4-15: Giao diện sau khi kết nối tất cả IP đến hệ thống 48
Hình 4-16: Giao diện thanh Port 49
Hình 4-17: Tạo INTERRUPT cho microblaze_0 49
Hình 4-18: Tạo INTERRUPT cho microblze_0 thành công 49
Hình 4-19: Kết nối Irq với microblze_0_INTERRUPT 50
Hình 4-20: Kết nối ngắt 50
Hình 4-21: Tạo ngắt delay 50
Hình 4-22: Cấu hình xps_intc_0 sau khi hoàn thành 50
Hình 4-23: Cấu hình delay ban đầu 51
Hình 4-24: Kết nối Capture Trig0 51
Hình 4-25: Cấu hình delay sau khi hoàn thành 51
Hình 4-26: Tạo kết nối Make External 51
Hình 4-27: Kết nối Make External thành công 52
Hình 4-28: Chọn dung lượng và địa chỉ bộ nhớ 52
Hình 4-29: Sau khi chọn dung lượng bộ nhớ cho các IP 52
Hình 4-30: File system.ucf 53
Hình 4-31: Kết nối bus 56
Hình 4-32: Biên dịch bitstream.bit 56
Hình 4-33: Cập nhật bitstream.bit 57
Hình 4-34: Lưu đồ chương trình chính 58
Hình 4-35: Xuất phần cứng qua phần mềm SDK 58
Hình 4-36: Mở phần mềm SDK 59
Hình 4-37: Chọn đường dẫn lưu thư mục SDK 59
Hình 4-38: Tạo phần mềm 59
Hình 4-39: Chọn loại board được hỗ trợ 60
Hình 4-40: Thông số 60
Hình 4-41: Tạo Project phần mềm 61
Hình 4-42: Project C 61
Hình 4-43: Tên Project và nền Project 62
Hình 4-44: Đối tượng 62
Hình 4-45: Phần mềm file C 63
Hình 4-46: Thư mục lưu file và tên file 63
Hình 4-47: Tạo file chỉ dẫn mã nguồn 64
Hình 4-48: Nơi lưu mã được biên dịch 64
Hình 4-49: Chèn lên file có sẵn 64
Hình 4-50: Program FPGA 65

Danh mục bảng

Bảng 2-1: Nguồn cung cấp cho Board Nexys 2 27
Bảng 2-2: Địa chỉ các chân I/O mở rộng 30
Bảng 2-3: Gán chân Bus dữ liệu và địa chỉ bộ nhớ 32

Bảng 3-1: Liệt kê các chân giao tiếp với các bộ kết nối 40

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT
THÀNH PHỐ HỒ CHÍ MINH
ĐƠN VỊ

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

Tp. HCM, ngày 9 tháng 3 năm 2015

THÔNG TIN KẾT QUẢ NGHIÊN CỨU

1. Thông tin chung:

- Tên đề tài: NGHIÊN CỨU GIAO TIẾP FPGA VỚI MODULE SIM VÀ VIẾT CHƯƠNG TRÌNH ĐIỀU KHIỂN CHO THIẾT BỊ TỪ XA
- Mã số: T2014 - 07
- Chủ nhiệm: GV.THS. TRƯƠNG THỊ BÍCH NGÀ
- Cơ quan chủ trì: Đại học Sư phạm Kỹ thuật Thành phố Hồ Chí Minh
- Thời gian thực hiện: **THÁNG 2/2014 ĐẾN THÁNG 12/2014**

2. Mục tiêu:

Nghiên cứu thiết kế và lập trình giao tiếp giữa FPGA , chip Spartan-3E, với module sim900 để điều khiển thiết bị từ xa bằng SMS.

3. Tính mới và sáng tạo:

Nghiên cứu lập trình cho chip Spartan-3E, chip FPGA của hãng Xilinx, giao tiếp với module sim900 bằng phương pháp lập trình nhúng. Dùng phần mềm EDK để thiết lập cấu hình và lập trình cho chip.

4. Kết quả nghiên cứu:

Đã nghiên cứu và lập trình thành công cho chip Spartan-3E giao tiếp với module sim900 để điều khiển các thiết bị từ xa bằng SMS.

5. Sản phẩm:

Chương trình ghi đĩa CD và mạch giao tiếp với các thiết bị thông qua cổng PMOD của kit Nexsy 2.

6. Hiệu quả, phương thức chuyển giao kết quả nghiên cứu và khả năng áp dụng:

Đây là một phương pháp dùng FPGA để điều khiển thay cho cách dùng vi điều khiển thông thường. Sẽ là một nguồn tài liệu dùng để tham khảo cho những ai muốn dùng FPGA để điều khiển.

P. Trưởng Đơn vị	**Chủ nhiệm đề tài**
(ký, họ và tên, đóng dấu)	*(ký, họ và tên)*
TS. Lê Chí Kiên	ThS.TRƯƠNG THỊ BÍCH NGÀ

RESULTS INFORMATION

1. General Information:

- Project title: RESEARCH COMMUNICATION A SIM MODULE WITH FPGA AND WRITING PROGRAM TO CONTROL REMOTE DEVICES
- Item Number: T2014 - 07
- Chairman: Msc. Truong Thi Bich Nga
- Responsible agencies: University of Technical Education Ho Chi Minh City
- Implementation period: 02/2014 to 12/2014

2. Objectives:

Design and programming communication between FPGA, Spartan-3E chip, with sim900 module to control remote devices by SMS.

3. Novelty and creativity:

Research programming for Spartan-3E chip, chip Xilinx FPGA vendors, communicate with sim900 module by means of embedded programming. EDK software for configuring and programming the chip.

4. Research results:

Studied, and successful programming for Spartan-3E chip communication with sim900 module to control the device remotely by SMS.

4. Products:

CD burning program and circuit communication with the device through the port of kit PMOD Nexsy 2.

6. Efficient method of transferring research results and applicability:

This is a method used to control FPGA instead of using conventional microcontroller. It is a resource used to refer to those who want to use FPGA to control.

Msc. Truong Thi Bich Nga

MỞ ĐẦU

Hiện nay công nghệ FPGA đang phát triển rất nhanh, những ứng dụng trong thực tế của FPGA rất rộng rãi, bao gồm: nhận dạng ảnh, nhận dạng tiếng nói, mật mã học..... Đặc biệt với khả năng tái lập trình, người sử dụng có thể thay đổi thiết kế của mình cả về phần cứng lẫn phần mềm.

Với trình độ hiện tại của nền công nghiệp nước ta, tài liệu và những kiến thức về FPGA vẫn còn ít nhưng nhu cầu ứng dụng ngày càng nhiều. Vì vậy tôi chọn ứng dụng công nghệ FPGA để điều khiển, và trong đề tài NCKH này tôi chọn đối tượng là điều khiển là module sim900 để điều khiển thiết bị bằng tin nhắn SMS.

TÌNH HÌNH NGHIÊN CỨU TRONG VÀ NGOÀI NƯỚC

Trong những năm gần đây, với sự phát triển của nghành công nghệ điện điện tử trong việc tạo ra những bộ vi xử lí và vi điều khiển, các hệ thống đo lường và điều khiển ngày càng thông minh hơn, giải quyết nhiều bài toán phức tạp hơn.

Bên cạnh đó công nghệ FPGA ra đời và phát triển rất nhanh. Tuy nhiên, ứng dụng dùng FPGA còn khá mới mẻ ở nước ta nói chung, và trường ta nói riêng, trong khi ưu điểm của FPGA là một thiết bị lập trình được, có số lượng cổng rất lớn, có thể thay thế cho cả một hệ thống gồm CPU, bộ điều khiển, bộ nhớ và các IP.

TÍNH CẤP THIẾT – MỤC TIÊU

Với trình độ hiện tại của nền công nghiệp nước ta, tài liệu và những kiến thức về FPGA vẫn còn ít nhưng nhu cầu ứng dụng ngày càng nhiều. Và các đề tài điều khiển thiết bị bằng tin nhắn SMS thường sử dụng vi điều khiển, nên tôi muốn rẽ sang một hướng mới hơn là ứng dụng công nghệ FPGA để điều khiển thiết bị bằng tin nhắn SMS.

CÁCH TIẾP CẬN

Tiến hành tìm tài liệu, đọc tài liệu hướng dẫn lập trình nhúng trên FPGA của hãng Xilnx, đó là phần mềm EDK. Khảo sát module sim900. Thiết kế module giao tiếp và lập trình cho FPGA.

Tiến hành thử nghiệm sản phẩm đã thi công, đánh giá sản phẩm.

PHƯƠNG PHÁP NGHIÊN CỨU

Đọc các tài liệu, phân tích các sản phẩm đã thực hiện, đánh giá ưu nhược, tiến hành thiết kế module và lập trình theo yêu cầu, thử nghiệm, thi công và hoàn thiện sản phẩm.

ĐỐI TƯỢNG VÀ PHẠM VI NGHIÊN CỨU

Nhóm nghiên cứu sẽ khảo sát kit Nexys 2 hãng Digilent, tìm hiểu phần mềm EDK của Xilinx.

Nghiên cứu thiết lập một hệ thống nhúng và phần mềm điều khiển các thiết bị ngoại vi dùng kit Nexys2 Spartan 3E. Trong đề tài tôi chỉ sẽ thiết kế 1 module giao tiếp với thiết bị gồm 3 bóng đèn.

NỘI DUNG NGHIÊN CỨU

Nội dung nghiên cứu được thực hiện gồm nghiên cứu và lập trình cho Spartan-3E trên kit Nexsy 2 giao tiếp với module sim900.

Tất cả các nội dung thực hiện được thể hiện qua 4 chương theo sau.

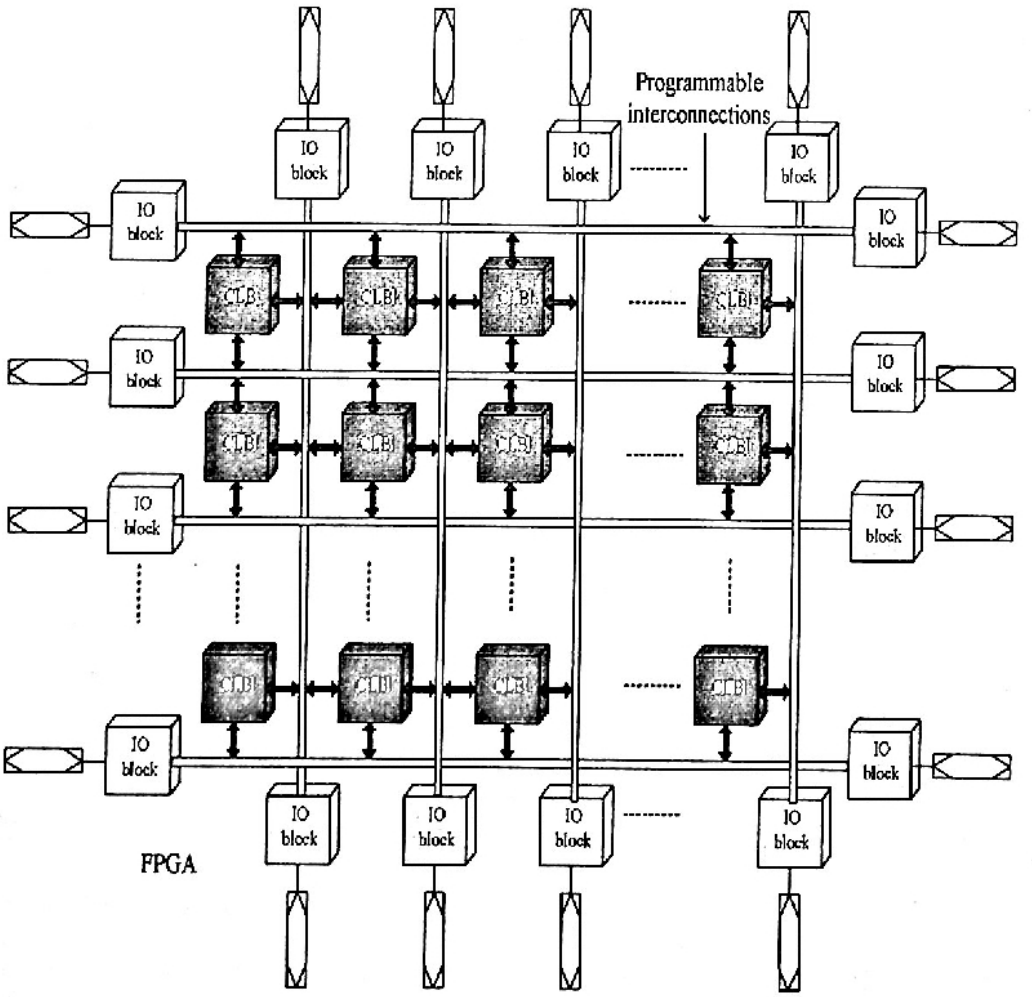
1.1 CÔNG NGHỆ FPGA

1.1.1 Cấu trúc FPGA

FPGA (Field Programmable Gate Array) không dùng mảng PAL/GAL. Các phần tử để tạo ra các hàm logic trong FPGA thường nhỏ hơn nhiều so với các thành phần trong CPLD.

Có 3 khối cơ bản trong FPGA là:

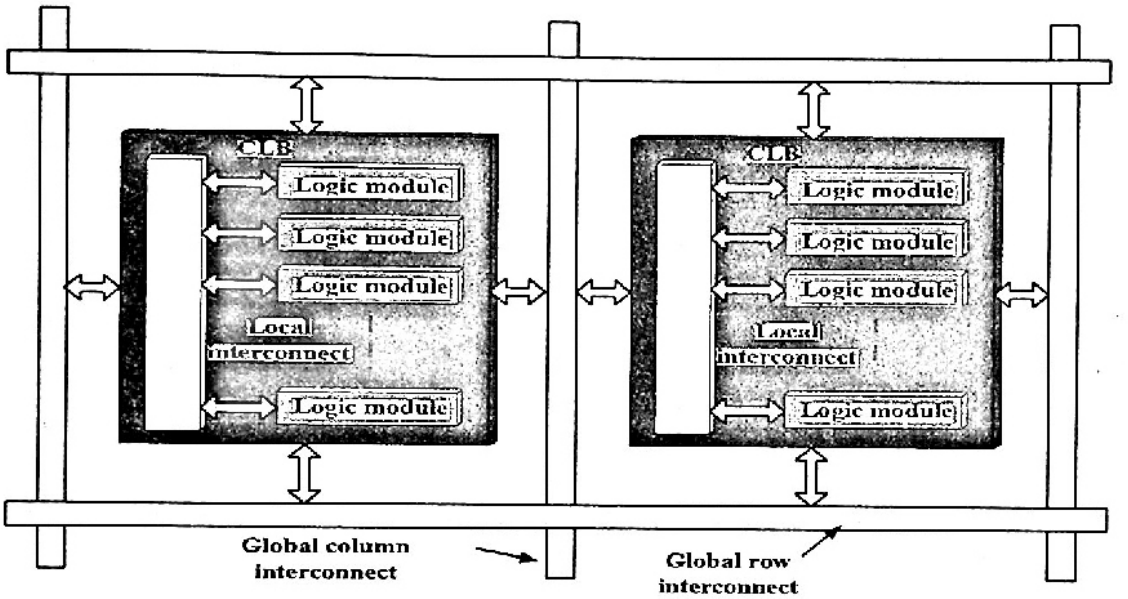
- ✓ Khối logic có thể định cấu hình CLB (Configurable Logic Block).
- ✓ Các kết nối bên trong.
- ✓ Các khối ngõ vào ra.



Hình 1-1 Cấu trúc FPGA

1.1.2 Khối logic có thể định cấu hình

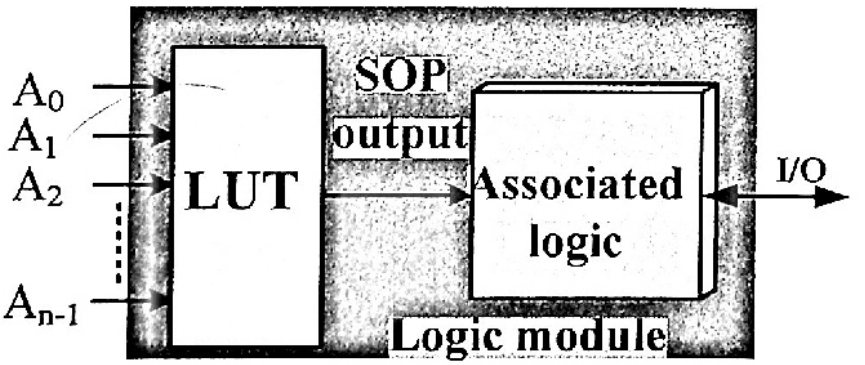
Khối logic của FPGA chứa một vài module logic khá nhỏ tương tự như macrocell trong CPLD. Các khối có thể lập trình hàng/cột toàn cục. Mỗi CLB được thiết lập từ nhiều module logic nhỏ hơn và các kết nối bên trong có thể lập trình cục bộ để kết nối các module logic khác với CLB.



Hình 1-2 Các khối CLB

1.1.3 Các module logic

Một module logc trong một khối logic của FPGA có thể được định cấu hình cho hàm logic tổ hợp, hàm logic thanh ghi hoặc cho cả 2. Flip flop là thành phần logic kết hợp và được dùng cho các hàm logic thanh ghi.



Hình 1-3 Sơ đồ khối của một module logic

LUT bao gồm một số các ô nhớ bằng 2^n trong đó n là số lượng các biến ngõ vào.

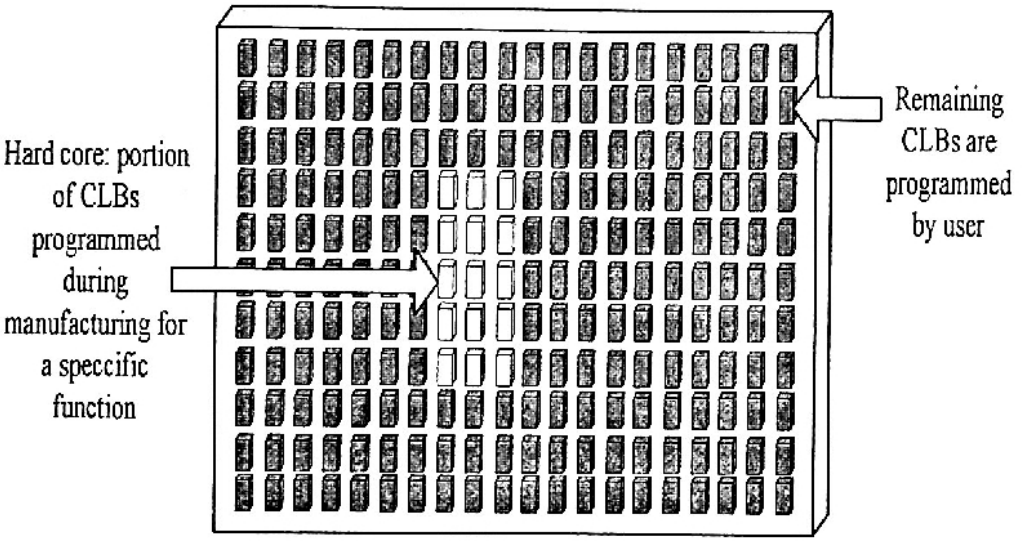
1.1.4 Các lõi của FPGA

Các FPGA giống như các phiến trắng mà người dùng có thể lập trình cho các thiết kế logic. Các FPGA tiện lợi khi mà nó chứa các mạch logic "lõi phần cứng". Một mạch logic lõi phần cứng là một phần logic trong FPGA được đặt vào bên trong bởi nhà chế tạo để cung cấp các chức năng đặc biệt và không thể lập trình lại.

Ưu điểm của dùng lõi phần cứng là: với cùng một yêu cầu thiết kế nếu dùng lõi phần cứng sẽ tốn ít tài nguyên của FPGA hơn so với cách sử dụng phần mềm để lập trình, những ưu điểm khác là không gian trên chip nhỏ hơn và thời gian thiết kế ngắn hơn.

Khuyết điểm là các thông số kỹ thuật cố định trong quá trình chế tạo và không có khả năng thay đổi về sau.

Các lõi phần cứng có các chức năng được sử dụng phổ biến trong các hệ thống như vi xử lý, mạch giao tiếp ngõ vào/ngõ ra và xử lý tín hiệu số. Có nhiều chức năng lõi phần cứng có thể lập trình trong FPGA.



Hình 1-4 Khái niệm chức năng lõi phần cứng trong FPGA

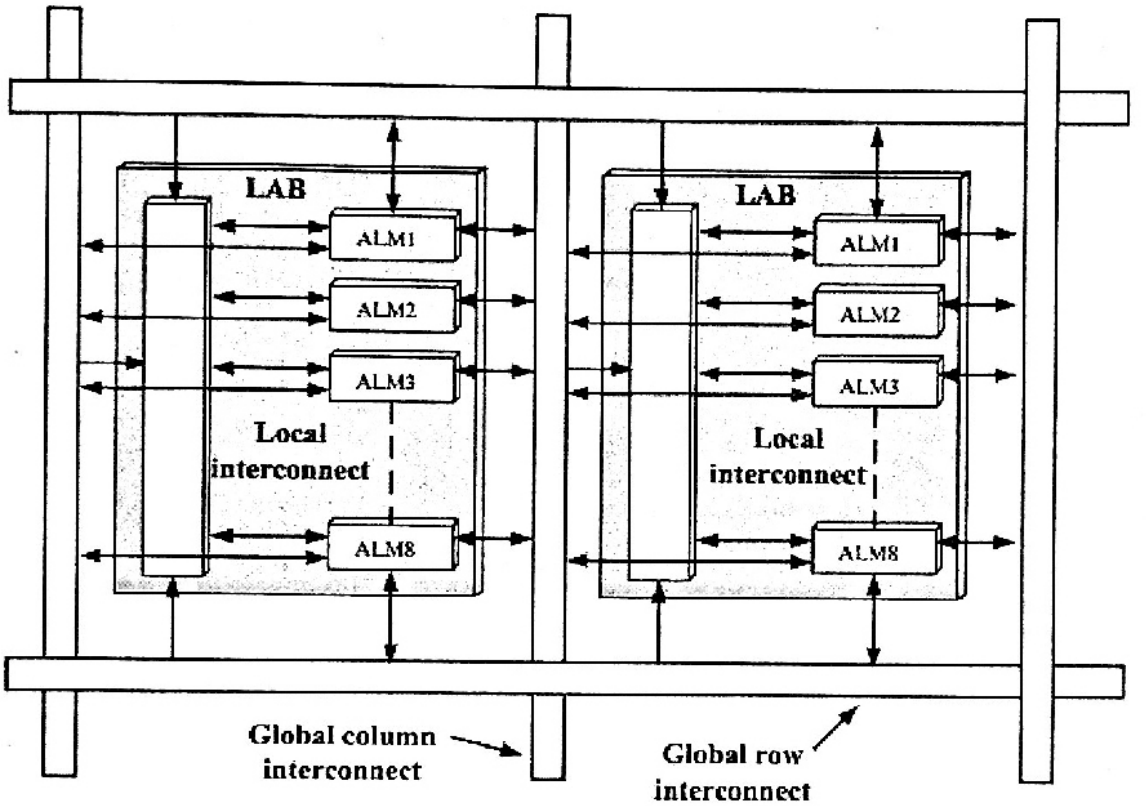
Việc thiết kế các lõi phần cứng thường được xây dựng bởi các nhà chế tạo FPGA và thuộc sở hữu của nhà chế tạo. Các thiết kế riêng bởi nhà chế tạo được đặt tên là Intellectual Property (IP) – sở hữu trí tuệ. Các công ty thường liệt kê các loại IP có hiệu lực trên các website. Nhiều IP kết hợp giữa phần mềm và phần cứng. Các FPGA chứa các vi xử lý tích hợp một trong hai hoặc cả hai lõi phần cứng và lõi phần mềm cùng nhiều chức năng khác đặt tên là Platform FPGA bởi vì chúng có thể được dùng điều khiển một hệ thống đầy đủ mà không cần thêm một thiết bị hỗ trợ nào.

1.2 FPGA CỦA ALTERA

Altera sản xuất ra nhiều họ FPGA bao gồm Statix II, Stratix, Cyclone và ACEX.

1.2.1 Khối mảng logic (LAB – Logic Array Block)

Sơ đồ khối của FPGA tổng quát đã được trình bày như hình 1-1; cấu trúc của Stratix II và các học Altera khác thì giống nhau. Chúng đều có cấu trúc loại LUT cho các module logic - được gọi là module logic thích nghi ALM (Adaptive Logic Module) được trình bay trong thiết bị tổng quát LAB. Mật độ được phân loại từ 2000 LAB cho đến 22000 LAB tùy thuộc vào các họ cụ thể và mỗi LAB có 8 ALM. Kích thước vỏ thay đổi từ 314 chân đến 1173 chân. Thiết bị yêu cầu sử dụng nguồn DC cung cấp từ 1,2V; 1,5V và 2,5V. Họ FPGA Stratix II sử dụng công nghệ SRAM.



Hình 1-5 Sơ đồ khối của cấu trúc LAB của Stratix II và ALM

Hình 1-5 trình bày sơ đồ khối của cấu trúc LAB của Stratix II. Mỗi LAB chứa 8 ALM, các LAB được liên kết với nhau thông qua các kết nối hang và cột bên trong. Các điểm kết nối cục bộ bên trong liên kết các ALM với mỗi LAB.

1.2.2 Module logic thích nghi ALM

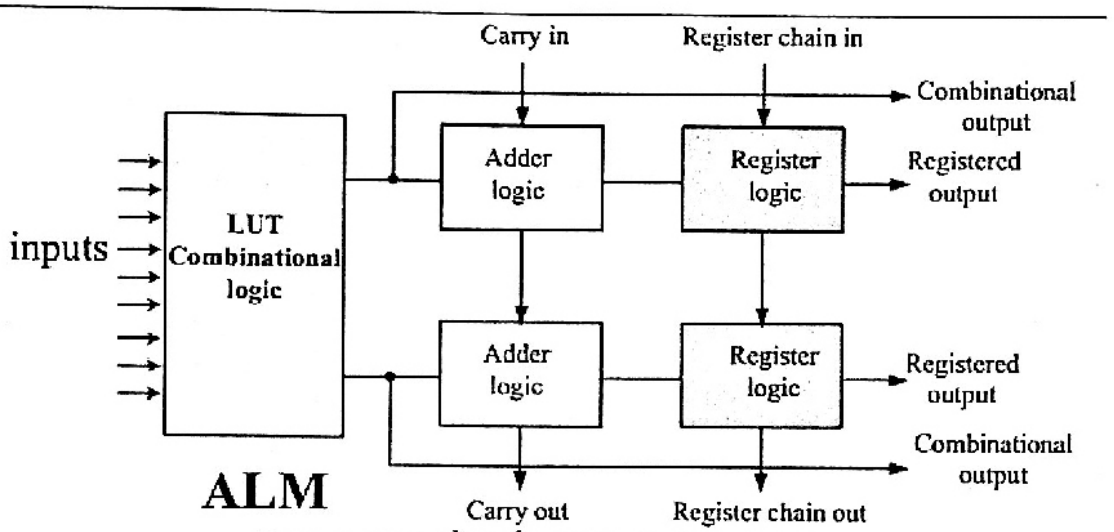
ALM là đơn vị thiết kế cơ bản trong FPGA Stratix II. Mỗi ALM chứa một thành phần tổ hợp logic dùng cấu trúc LUT và mạch logic kết hợp có thể được lập trình cho 2 ngõ ra logic tổ hợp hoặc hai ngõ ra thanh ghi dịch. Bên cạnh đó, ALM có mạch cộng logic, các flip flop và các mạch logic khác – cho phép thực hiện chức năng tính toán số học, chức năng đếm và thanh ghi dịch. Sơ đồ khối của ALM của Stratix II được trình bày như hình 1-6.

Hoạt động của ALM:

Một ALM có thể được lập trình cho ra nhiều kiểu hoạt động như sau:

- Kiểu hoạt động bình thường.
- Kiểu hoạt động LUT mở rộng
- Kiểu tính toán số học.
- Kiểu tính toán số học dùng chung.

Ngoài 4 kiểu hoạt động thì ALM có thể được dùng như là 1 chuỗi thanh ghi để xây dựng bộ đếm và thanh ghi dịch.

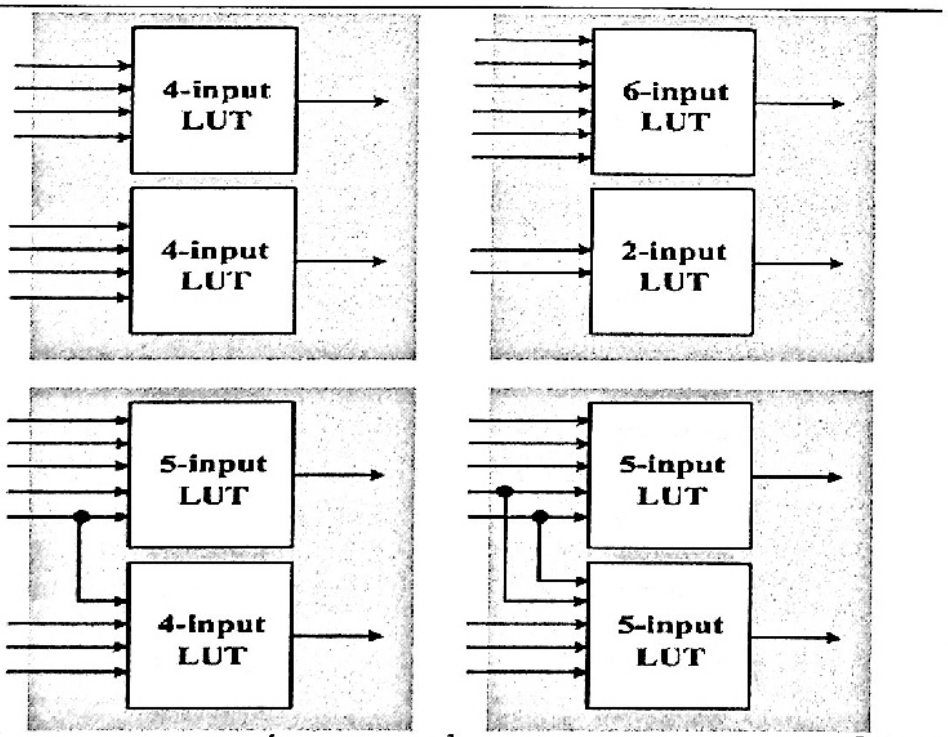


Hình 1-6 Sơ đồ khối ALM của Stratix II.

1.2.2.1Kiểu hoạt động bình thường

Được sử dụng đầu tiên để tạo các hàm logic tổ hợp. Một ALM có thể thực hiện một hoặc hai hàm ngõ ra tổ hợp với hai LUT của nó. Ví dụ về 4 cấu hình LUT được minh họa như hình 1-7.

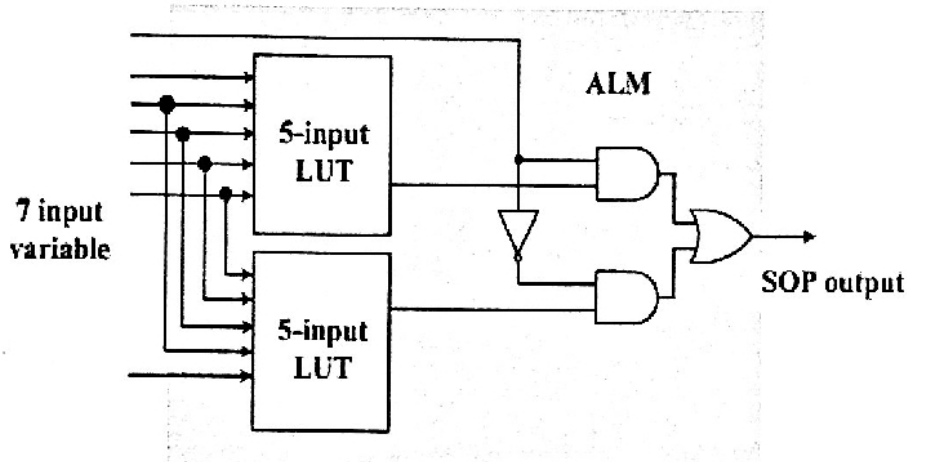
Hai hàm SOP - mỗi hàm có 4 biến hoặc ít hơn – có thể được thực hiện trong một ALM mà không cần dùng các ngõ vào chia sẻ.



Hình 1-7 Các cấu hình có thể có của LUT trong ALM ở kiểu bình thường.

1.2.2.2 Kiểu hoạt động LUT mở rộng

Cho phép mở rộng hàm lên đến 7 biến được minh họa ở hình 1-8. Mạch điện AND – OR với các ngõ vào đảo là một ví dụ đơn giản của mạch dồn kênh. Mạch dồn kênh là một phần của mạch logic dùng riêng trong ALM.



Hình 1-8 Mở rộng ALM để tạo ra hàm SOP 7 biến trong kiểu LUT mở rộng.

1.2.3 Các chức năng tích hợp

Sơ đồ khối tổng quát của FPGA Stratix II được trình bày ở hình 1-8. FPGA chứa các thành phần bộ nhớ và chức năng xử lý tín hiệu số DSP. Chức năng của DSP như các mạch lọc số thường được sử dụng nhiều trong các hệ thống. Khi quan sát sơ đồ khối, các khối tích hợp bên trong được sắp xếp ở khắp nơi trong ma trận kết nối bên trong của FPGA và các thành phần từ ngõ vào/ra được đặt xung quanh chu vi FPGA.

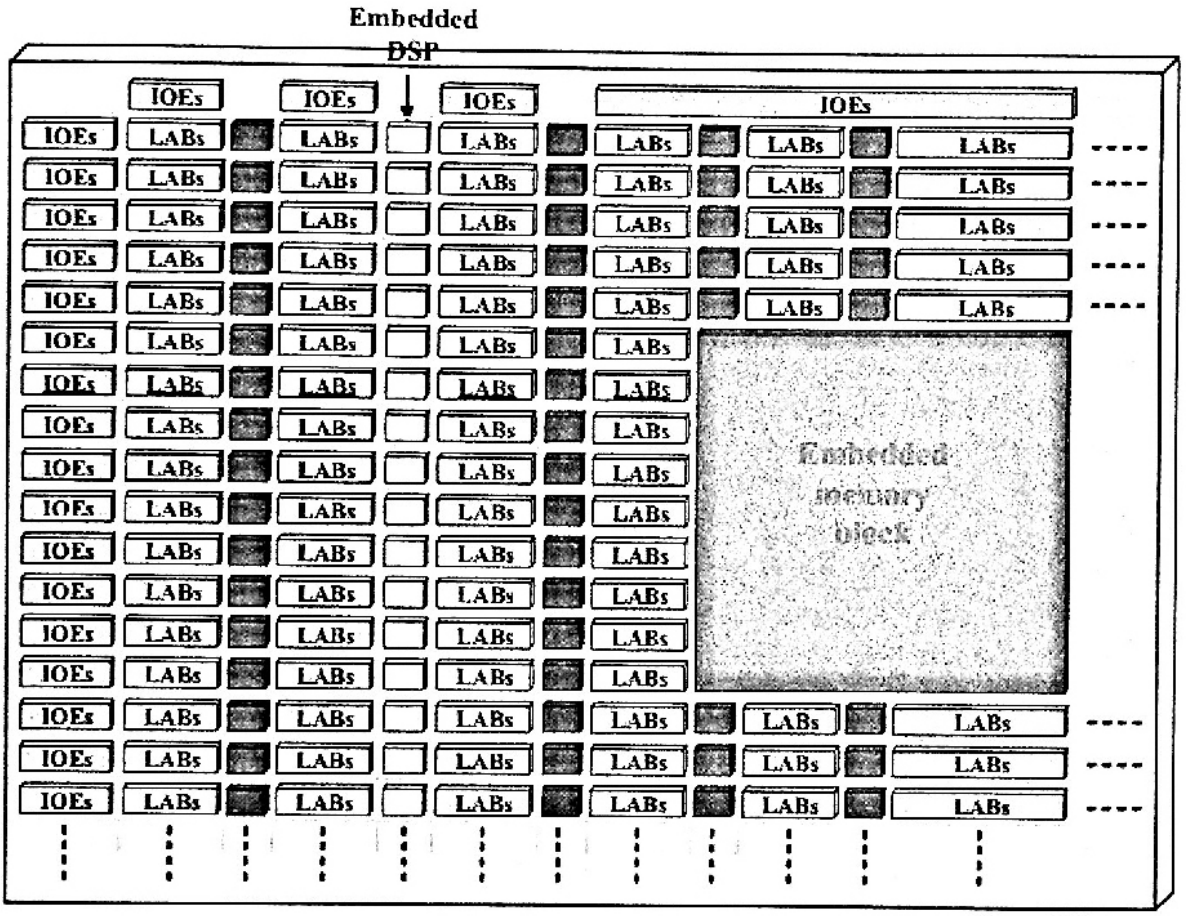
1.3 FPGA CỦA HÃNG XILINX

1.3.1 Các khối logic có thể định cấu hình CLB (Configurable Logic Block)

Vùng logic định cấu hình của hầu hết các FPGA họ Xilinx được chia thành nhiều khối logic có thể định cấu hình CLB với mỗi CLB chứa nhiều đơn vị logic cơ bản được là các tế bào logic (logic cell – LC).

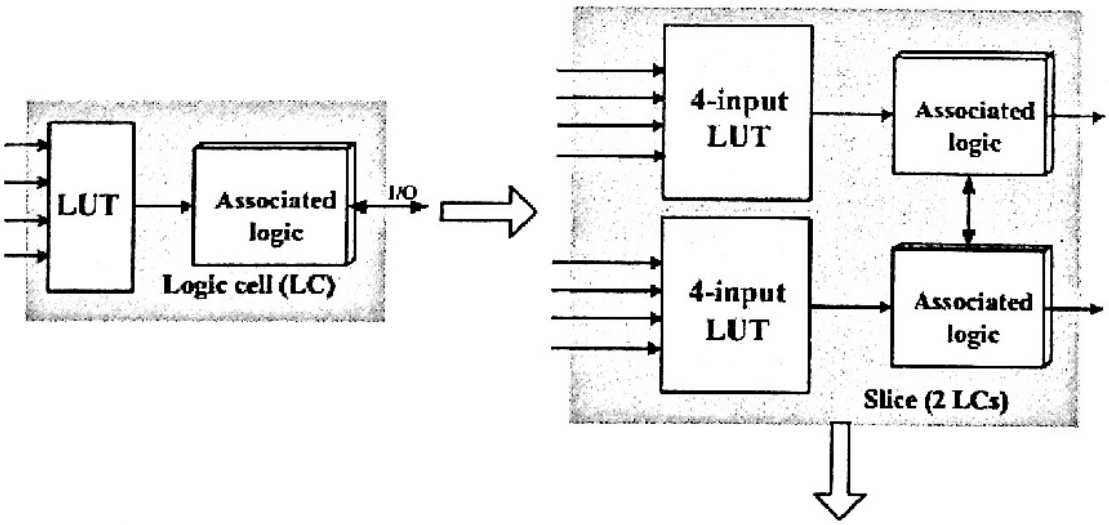
Mỗi tế bào logic LC sử dụng mạch logic LUT truyền thống có 4 ngõ vào và thêm mạch logic cộng và một flip flop.

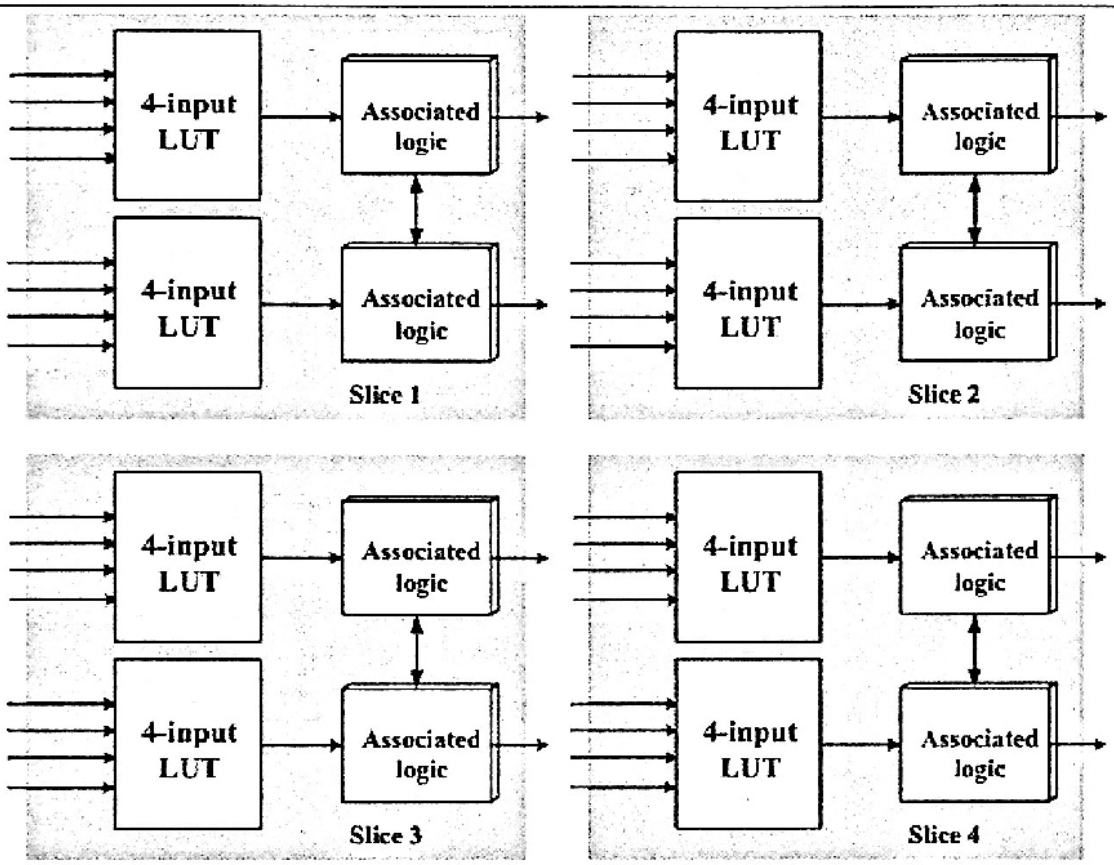
Một LUT có 4 ngõ vào có thể tạo ra hàm SOP chứa 16 thành phần tích. Hai tế bào logic LC giống nhau được gọi là slice (lát mỏng).



Hình 1-9 Sơ đồ khối của FPGA Stratix II.

Hình 1-9 minh họa các cấp logic định cấu hình từ tế bào logic cho đến CLB. Mật độ tích hợp nằm trong khoảng từ 2000 đến 74000 tế bào logic LC trong một thiết bị Virtex đơn.





Hình 1-10 Minh họa các cấp logic định cấu hình từ tế bào logic cho đến CLB.

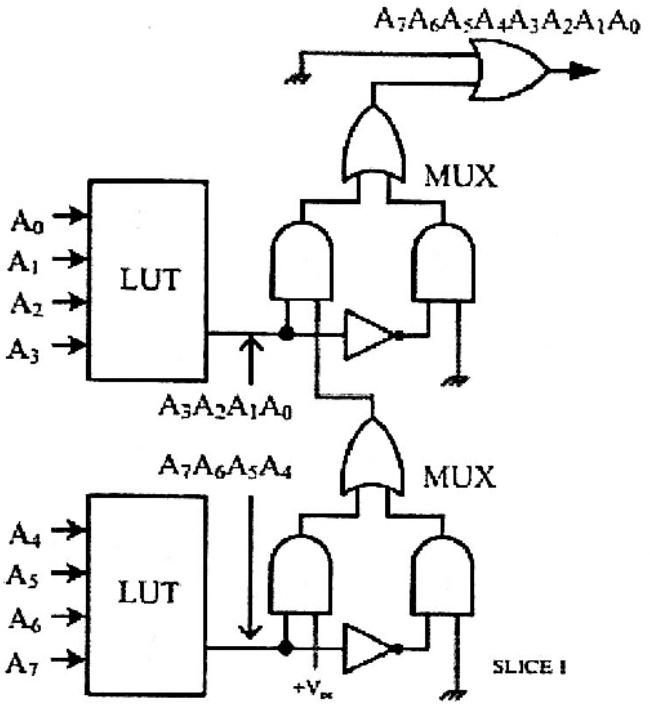
1.3.2 Chuỗi liên tiếp SOP

Slice đơn giản (hai tế bào logic LC) với logic chuỗi liên tiếp được tình bày ở hình 1-11. Có mạch đa hợp (MUX) dành riêng nằm trong mạch logic kết hợp của mỗi LC - được dùng trong chuỗi liên tiếp và một cổng OR dành riêng nằm trong slice.

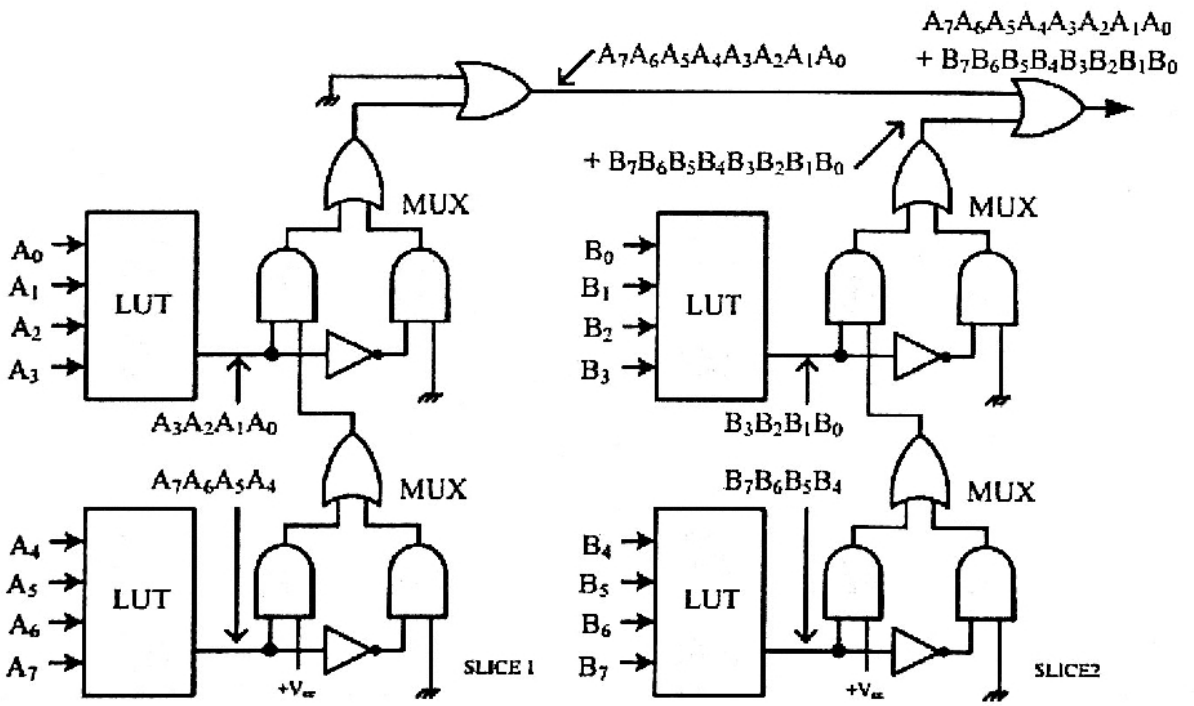
Hình 1-11a được trình bày ví dụ cách mà một slice trong CLB có thể định được cấu hình như là một cổng AND để tạo ra thành phần tích 8 biến. Hai slice có thể được định cấu hình để tạo ra một hàm SOP với 2 thành phần tích 8 biến được trình bày ở hình 1-11b.

Toàn bộ CLB của 4 slice thể được định cấu hình thành một chuỗi liên tiếp để tạo ra một hàm SOP với 4 thành phần tích có 8 biến được trình bày như hình 1-11c. Biểu thức SOP dài hơn nữa có thể được thực hiện dùng thêm các CLB mở rộng.

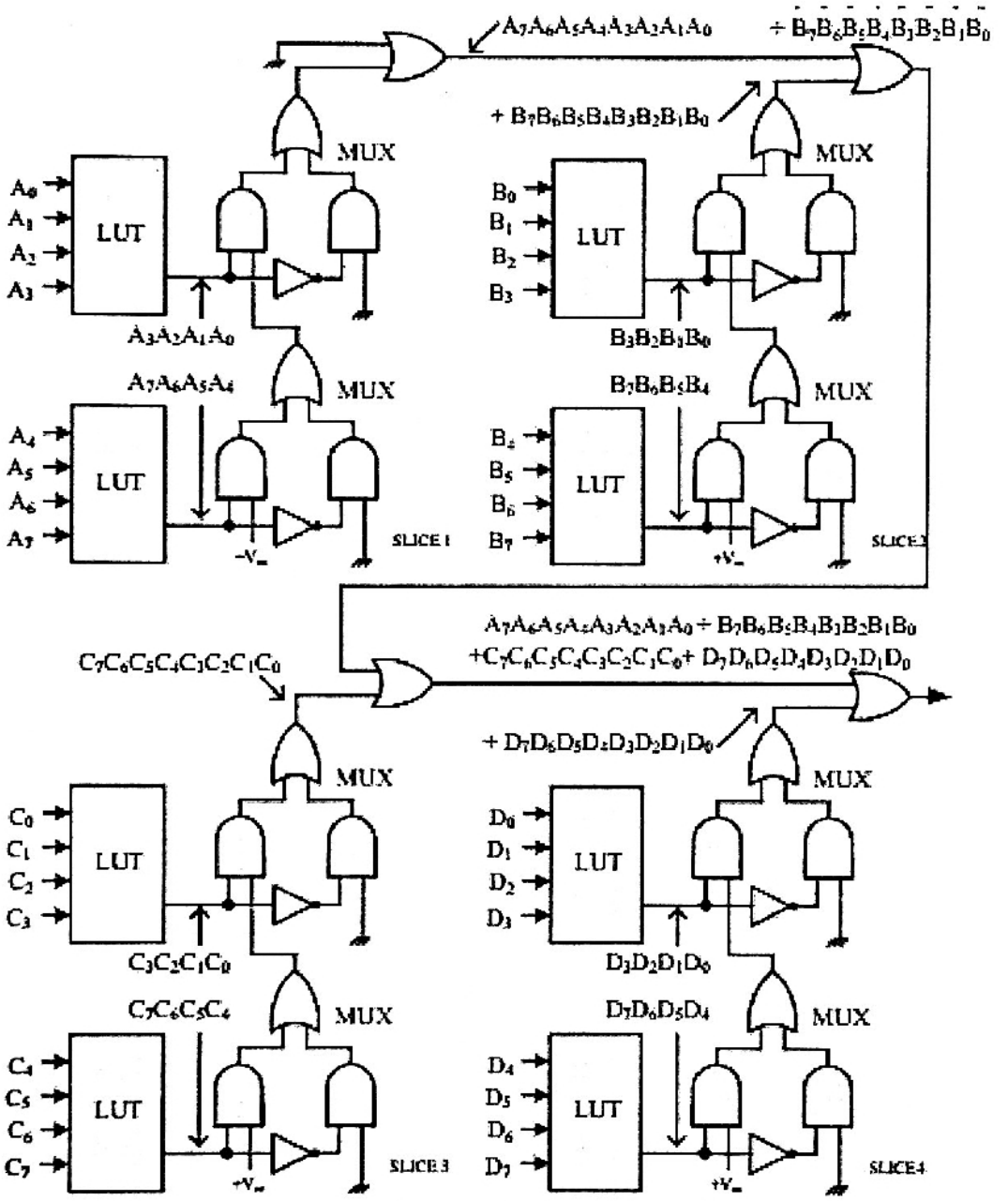
TH
SKC 005640


A7A6A5A4A3A2A1A0
MUX
A0
A1
A2
A3
LUT
A3A2A1A0
A7A6A5A4
MUX
A4
A5
A6
A7
LUT
+Vcc
SLICE 1

(a)


A7A6A5A4A3A2A1A0
A7A6A5A4A3A2A1A0
+ B7B6B5B4B3B2B1B0
+ B7B6B5B4B3B2B1B0
MUX
MUX
A0
A1
A2
A3
LUT
B0
B1
B2
B3
LUT
A3A2A1A0
B3B2B1B0
A7A6A5A4
MUX
B7B6B5B4
MUX
A4
A5
A6
A7
LUT
B4
B5
B6
B7
LUT
+Vcc
SLICE 1
+Vcc
SLICE2

(b)



Hình 1-11 Cách dùng chuỗi nối tiếp để mở rộng biểu thức SOP

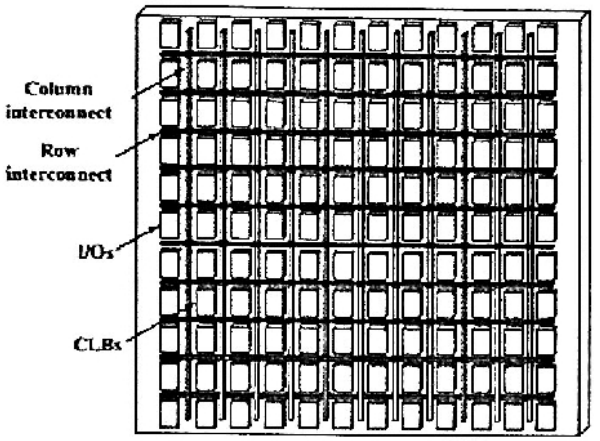
1.3.3 Cấu trúc FPGA truyền thông và cấu trúc ASMBL.

1.3.3.1 Cấu trúc truyền thông

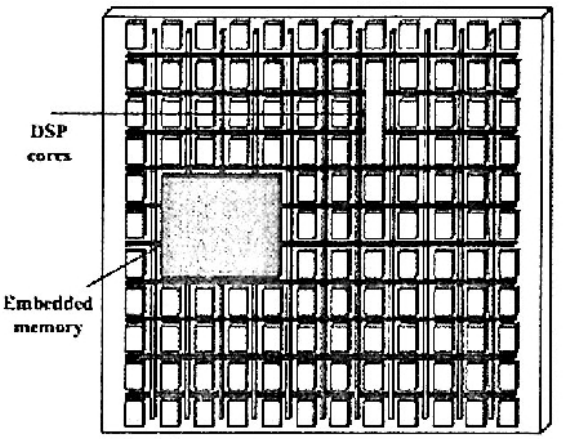
Cấu trúc FPGA truyền thông xuất hiện như là một mảng của các khối logic (CLB hoặc LAB) được bao bọc xung quanh bởi các tế bào ngõ vào/ra có thể định cấu hình. Số CLB trong FPGA tùy thuộc vào số lượng các phần tử IO – có thể đặt xung quanh.

Khi lõi IP như DSP và bộ nhớ tích hợp bên trong khi được yêu cầu thì một lượng logic định cấu hình phải mất đi và tại một vị trí được thay thế bằng IO nếu có yêu cầu.

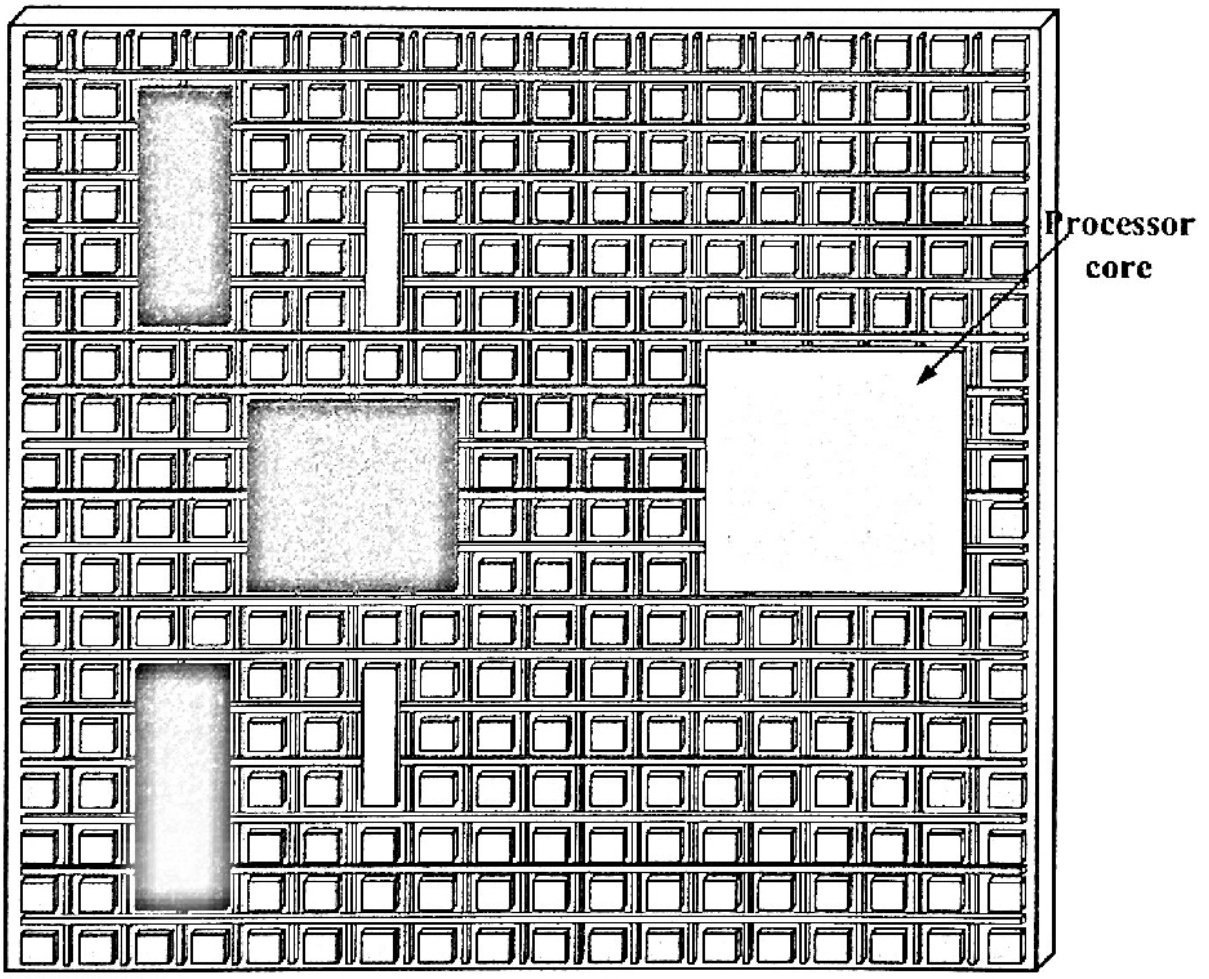
Khi nhiều lõi IP được thêm vào thì kích thước vật lý của FPGA phải tăng lên để đảm bảo số lượng logic cấu hình cần thiết và tăng thêm số lương IO. Khái niệm này được minh họa bằng hình 1-12.



(a) FPGA với logic định hình đầy đủ.



(b) FPGA cùng kích thước với bộ nhớ và lõi IP (DSP) nên có ít CLB hơn.



(c) FPGA có nhiều bộ nhớ, thêm lõi DSP và lõi vi xử lý sẽ yêu cầu kích thước lớn hơn.

Hình 1-12 Tích hợp nhiều chức năng IP

Cấu hình trong FPGA càng phức tạp thì càng dùng nhiều IO. Mối liên hệ ràng buộc giữa logic và IO sẽ dẫn đến tăng kích thước chip và tăng giá thành. Ngoài ra một vấn đề khác với FPGA platform là khi thêm các chức năng lõi IP tích hợp

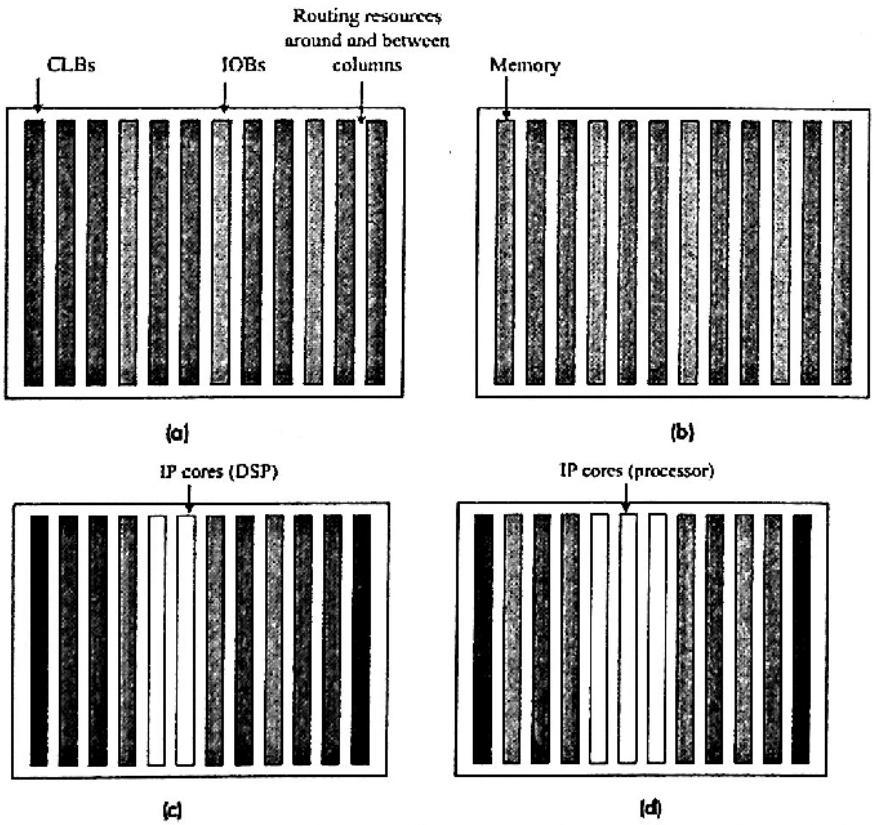
bên trong nếu có yêu cầu thì phải thiết kế lại thành phần chính hoặc thiết kế lại một phần trong cách bố trí chip (layout) nếu được yêu cầu sẽ tăng thêm giá thành.

1.3.3.2Cấu trúc ASMBL

Xilinx đã xây dựng một phương pháp mềm dẻo cho FPGA platform ở chip Virtex II Pro X để khắc phục một vài hạn chế xuất hiện trong cấu trúc truyền thông. Cấu trúc ASMBL là cấu trúc sử dụng cột thay vì dùng cấu trúc hàng cột/dọc. Các IO được đặt rải rác khắp nơi tốt hơn là đặt xung quanh, dẫn đến số lượng IO của nó tăng mà không cần làm tăng kích thước chip. Mỗi cột về cơ bản là một dải logic có thể được thay thế bằng dải logic khác mà không cần thiết kế lại cách bố trí chip. Các ví dụ về các loại của các dải logic là các khối logic định cấu hình CLB, khối IO, bộ nhớ và các lõi phần cứng và phần mềm như DSP và vi xử lý.

Số lượng khác nhau của mỗi loại dải logic có thể được trộn lại để tương thích với các yêu cầu ứng dụng riêng biệt. Ví dụ, trong cấu hình đơn giản nhất thì có thể pha trộn các dải CLB và các dải khối IO được minh họa như hình 1-13a. Nhiều hoặc ít hơn của cả 2 cũng có thể được sử dụng tùy thuộc vào các yêu cầu.

Nếu cần nhiều bộ nhớ thì một hoặc nhiều dải CLB có thể được thay thế như hình 1-13b. Nếu vùng riêng biệt trong ứng dụng là xử lý tín hiệu số thì có thể thêm vào các lõi IP DSP trộn với bộ nhớ như hình 1-13c. Hình 1-13d trình bày các lõi vi xử lý được thêm vào.

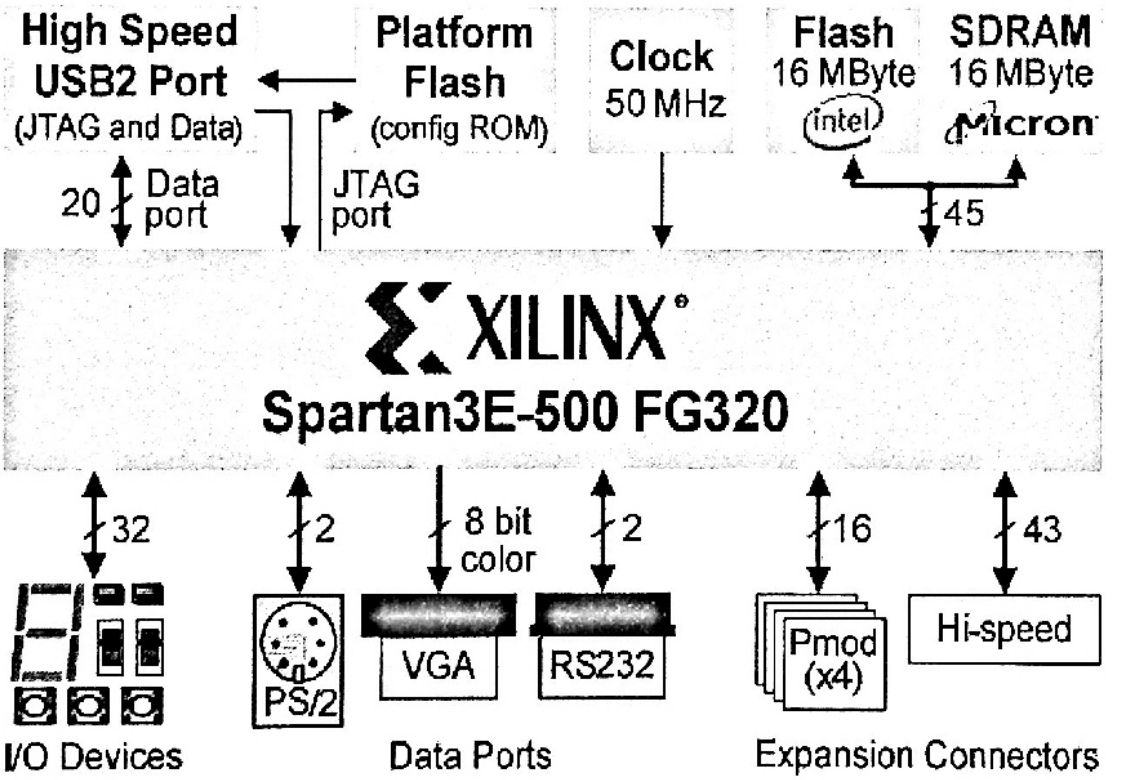


Hình 1-13 Minh họa cấu trúc ASMBL của FPGA platform.

GIỚI THIỆU KIT DILIGENT NEXYS 2 SPARTAN-3E

2.1.Tổng quan về Board Nexys 2

Board Nexys2 đã được hoàn thiện, chip sử dụng là Xilinx Spartan3E FPGA. Tích hợp cổng USB2 tốc độ cao, 16Mbytes bộ nhớ RAM và ROM, nhiều thiết bị I /O và các cổng làm cho nó trở thành một nền tảng lý tưởng cho các hệ thống kỹ thuật số của tất cả các loại, bao gồm cả hệ thống bộ vi xử lý nhúng dựa trên MicroBlaze Xilinx. Cổng USB2 cung cấp nguồn và giao diện lập trình, do đó Board Nexys 2 có thể sử dụng với một máy tính xách tay để tạo thành một trạm thiết kế thực sự tiện dụng.



Hình 2-1 Sơ đồ khối và đặc điểm của Nexys 2

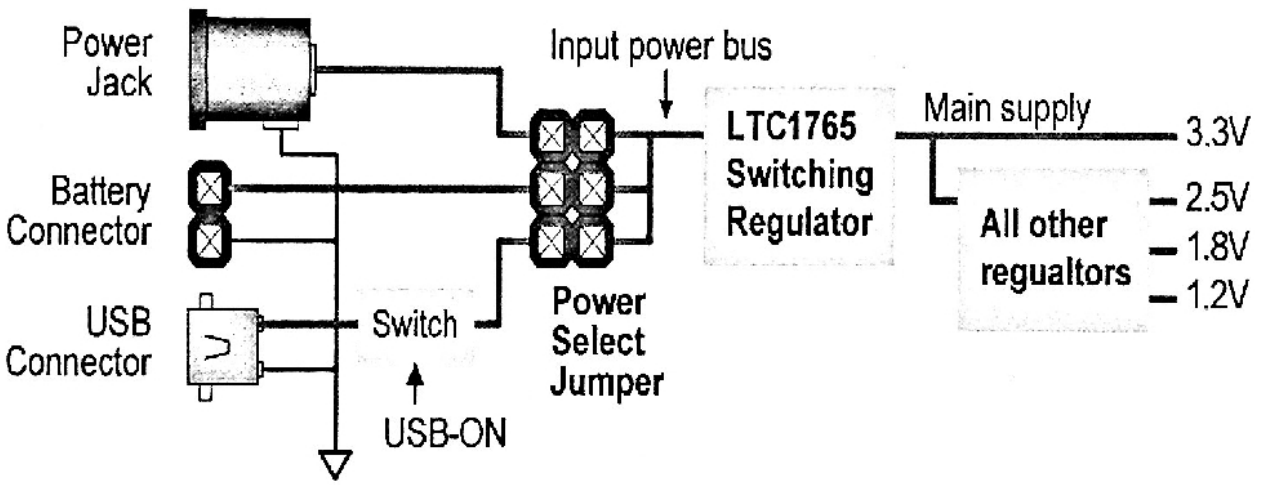
Các thành phần của Board Nexys 2 gồm:

- 500K-gate Xilinx Spartan 3E FPGA
- USB2 – cơ sở để FPGA cấu hình và truyền dữ liệu tốc độ cao (sử dụng bộ phần mềm Adept miễn phí)
- USB2 – nguồn cung cấp (pin và phích cắm cũng có thể được sử dụng)
- 16MB của Micron PSDRAM &16MB của Intel StrataFlash ROM
- Xilinx Platform Flash cho cấu hình FPGA không biến đổi
- Có phím gạt nguồn cung cấp hữu dụng (tốt cho các ứng dụng sử dụng năng lượng riêng)

- Bộ tạo dao động 50Mhz với 1 một ổ cắm cho dao động thứ 2.
- PS2 port (for mouse/keyboard).
- 8-bit VGA port.
- 60 I/O định tuyến cho kết nối mở rộng.
- Các ngõ ra đa dụng (GPIO) bao gồm 8 LED đơn, 4 nút nhấn, 8 công tắc 4 led 7 đoạn.

2.2 Nguồn cung cấp cho Board

Nguồn cung cấp cho Board Nexys 2 có thể được truyền từ một dây USB, từ một nguồn dương 5VDC – 15VDC, phích cắm nguồn 2.1mm, hoặc từ một bộ pin. Khối Power Select Jumper dùng để lựa chọn nguồn cung cấp cho Board. Hệ mạch USB luôn được cấp điện từ dây cáp USB, nếu không có cáp USB được cắm vào thì hệ mạch USB sẽ không được cung cấp điện.



Hình 2-2 Sơ đồ khối nguồn cung cấp Nexys 2

Đầu vào chính của nguồn được ổn áp ở điện áp 3.3V cung cấp tất cả dòng điện mà Board yêu cầu. Một số thiết bị yêu cầu 2.2V, 1.8V và 1.2V cung cấp thêm vào nguồn cung cấp chính là 3.3V, và các nguồn được cung cấp thêm này được tạo ra từ sự điều chỉnh nguồn cung cấp chính 3.3V. Các nguồn cung cấp chính được tạo ra bởi các máy điều chỉnh chuyển mạch hiệu suất cao từ công nghệ tuyến tính. Các máy điều chỉnh này không chỉ sử dụng hiệu quả nguồn USB, chúng cũng cho phép Nexys 2 chạy từ bộ pin trong thời gian dài.

Dòng tổng của Board phụ thuộc vào cấu hình của FPGA, tần số xung Clock và các kết nối bên ngoài. Trong mạch thử nghiệm với khoảng 20K cổng định tuyến, tần số 50MHz, và tất cả các đèn LED đều sáng, khoảng 200mA hiện tại được rút ra từ nguồn cung cấp 1.2V, 50mA từ nguồn cung cấp 2.5V, 100mA từ nguồn cung cấp 3.3V. Yêu cầu về dòng của mạch sẽ tăng nếu mạch mạch lớn hơn được cấu hình trong FPGA và nếu các thiết bị ngoại vi được gắn vào. Bảng 123 tóm tắt các thông số cung cấp điện.

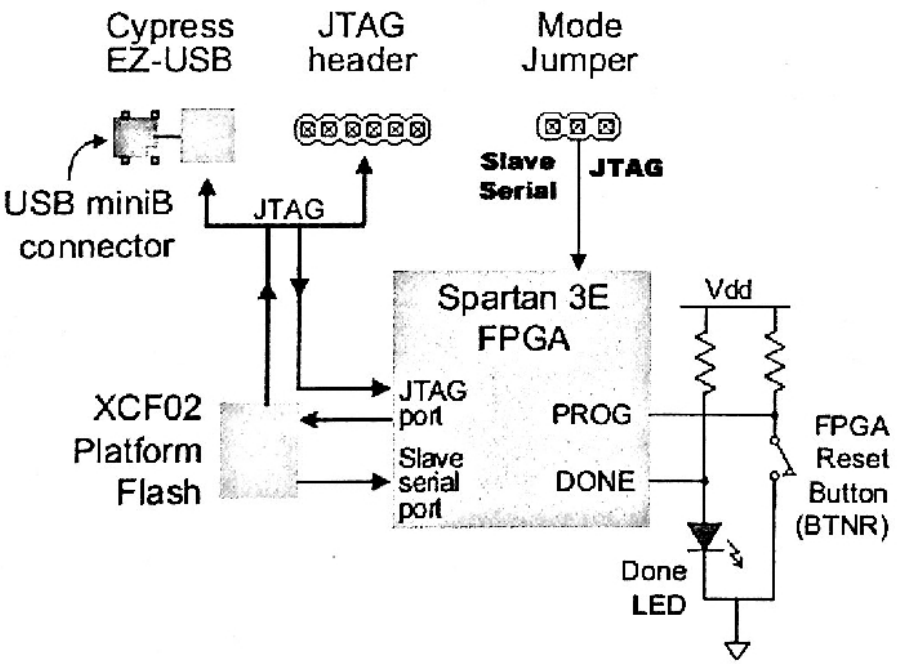
Supply	Device	Amps (max/typ)
3.3V main	IC6: LTC1765	3A/100mA
2.5V FPGA	IC7: LTC3417	1.4A/50mA
1.2V FPGA	IC7: LTC3417	1.4A/200mA
1.8V SRAM	IC8: LTC1844	150mA/90mA
3.3V USB	IC5: LTC1844	150mA/60mA

Bảng 2-1 Nguồn cung cấp cho Board Nexys 2

2.3 Cấu hình FPGA và Platform Flash

FPGA FPGA trên Board Nexys2 phải được cấu hình (hay được lập trình) của người sử dụng trước khi nó có thể thực hiện bất kỳ chức năng nào. Trong quá trình cấu hình, một tập tin "bit" được truyền vào các ô nhớ trong FPGA để xác định các chức năng logic và mạch kết nối. Phần mềm ISE / WebPack CAD miễn phí từ Xilinx có thể được sử dụng để tạo ra các tập tin bit từ VHDL, Verilog, hoặc các tập tin mã nguồn dựa trên sơ đồ.

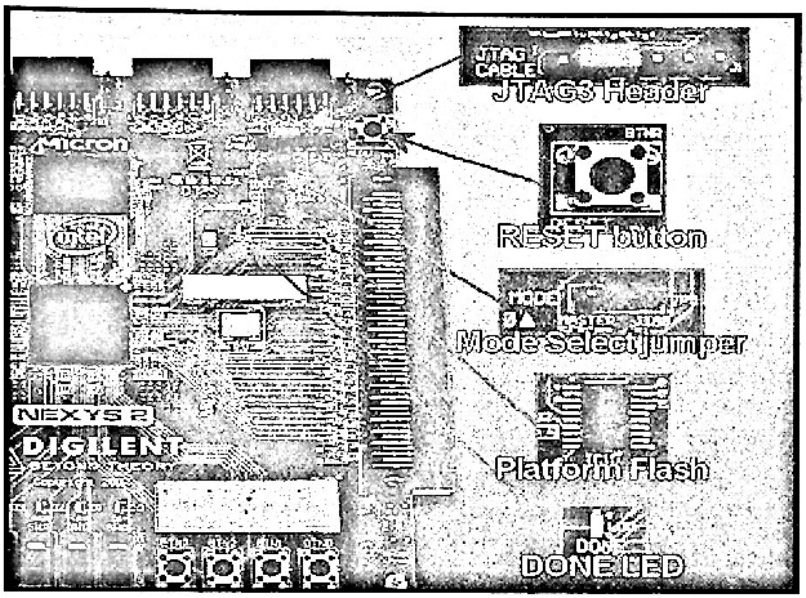
FPGA có thể được lập trình theo hai cách: trực tiếp từ một máy tính bằng cách sử dụng cổng USB trên Board, và từ một Platform Flash ROM (Flash ROM cũng là người dùng có thể lập trình thông qua cổng USB) trên Board. Một chân nối trên Board Nexys2 xác định nguồn (PC hoặc ROM) FPGA sẽ sử dụng để tải cấu hình của nó. FPGA sẽ tự động tải một cấu hình từ ROM Flash Platform khi bật nguồn nếu cấu hình chế độ chân nối được thiết lập "Master Serial". Nếu chế độ chân nối được thiết lập "JTAG", FPGA sẽ đợi chương trình từ máy PC (thông qua cáp USB).



Hình 2-3 Mạch lập trình Nexys 2

Có thể dùng phần mềm Adept miễn phí sẵn có trên máy tính hay phần mềm EDK để có thể được sử dụng để cấu hình FPGA và Flash Platform phù hợp với bất kỳ tập tin được lưu trữ trên máy tính. Adept sử dụng cáp USB để chuyển một tập tin .bit hay EDK chuyển một

tập tin .elf được chọn từ máy PC tới FPGA hoặc Platform Flash ROM. Sau khi FPGA được cấu hình, nó sẽ giữ nguyên cho đến khi nó được thiết lập lại do ảnh hưởng của nguồn hoặc nút reset FPGA (BTNR) được nhấn. ROM Flash Platform sẽ giữ lại một tập tin bit hay .elf cho đến khi nó được lập trình lại, bất chấp ảnh hưởng của nguồn.



Hình 2-4 Mạch lập trình Board Nexys 2

Để lập trình cho Board Nexys2 sử dụng Adept hay EDK, cắm cáp USB vào Board (nếu nguồn USB sẽ không được sử dụng, gắn một nguồn cung cấp phù hợp để các jack cắm điện hoặc pin kết nối trên Board, và thiết lập chuyển tới "wall" hoặc "bat"). Khởi động phần mềm Adept hay EDK, và chờ đợi cho FPGA và Flash Platform ROM để được chấp nhận. Sử dụng chức năng duyệt để liên kết các tập tin .bit hay .elf với FPGA, hoặc các tập tin .mcs với ROM Flash Platform. Nhấp chuột phải vào thiết bị được lập trình, và chọn chức năng "lập trình". Các tập tin cấu hình sẽ được gửi đến FPGA hoặc Flash Platform, và phần mềm sẽ cho biết lập trình đã hoàn thành. Các LED sẽ sáng sau khi FPGA đã được cấu hình thành công. Để biết thêm thông tin về cách sử dụng Adept, chúng ta có thể xem tài liệu Adept có sẵn tại trang web Digilent.

Board Nexys2 cũng có thể được lập trình sử dụng phần mềm iMPACT Xilinx bằng cách kết nối một cáp lập trình phù hợp với các đầu JTAG. Cáp JTAG3 Digilent hoặc bất kỳ cáp nào của Xilinx đều có thể được sử dụng.

Một cấu hình Demo được nạp vào Flash Platform trên bảng Nexys2 trong quá trình chế tạo. Cấu hình đó, cũng có sẵn trên trang web Digilent, có thể được sử dụng để kiểm tra tất cả các thiết bị và mạch trên bảng Nexys2.

2.4 Clock

Board Nexys2 bao gồm một bộ dao động 50MHz và một ổ cắm cho một dao động thứ hai. Tín hiệu clock từ các bộ dao động kết nối với toàn bộ clock đầu vào các chân trên FPGA bởi vậy chúng có thể điều khiển đồng bộ clock các khối đang hoạt động trong FPGA. Bộ tổng hợp clock (gọi là DLL, hoặc vòng khóa trì hoãn) cung cấp khả năng quản lý clock bao gồm tăng gấp đôi hoặc nhân bốn tần số đầu vào, phân chia tần số đầu vào bất kỳ số nguyên, và xác định giai đoạn chính xác và các mối liên hệ trễ giữa tín hiệu clock khác nhau.

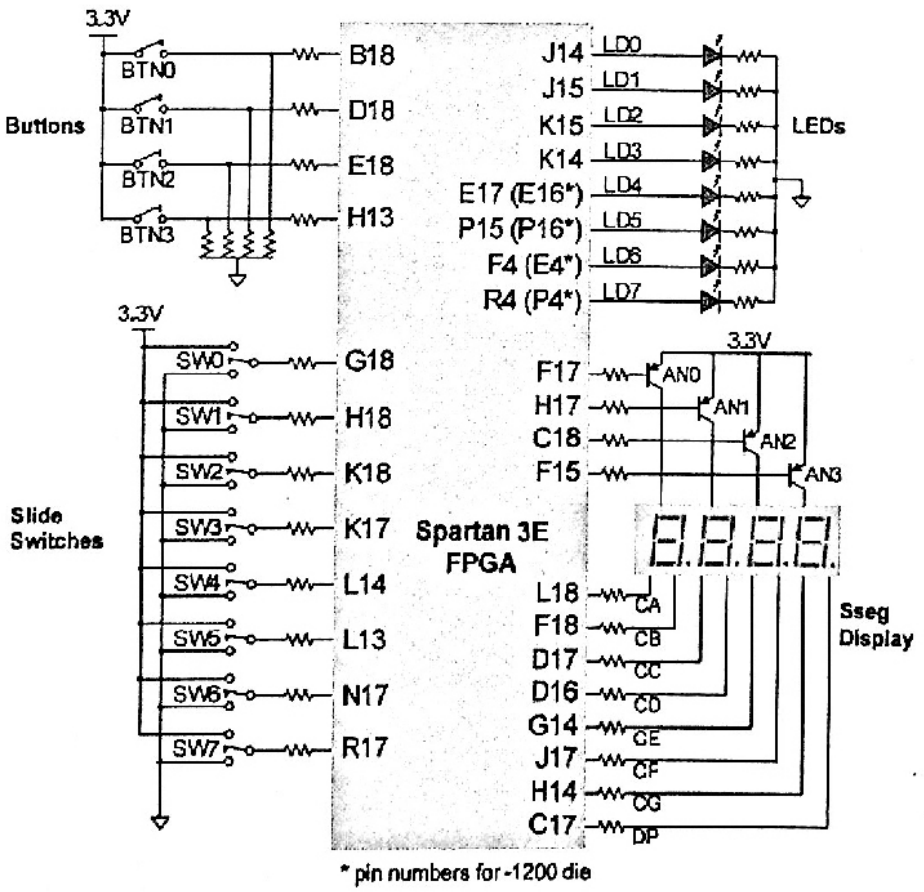
Những I / O sử dụng

Board Nexys2 bao gồm một số các thiết bị đầu vào, thiết bị đầu ra, và cổng dữ liệu, cho phép nhiều thiết kế được thực hiện mà không có sự cần bất kỳ thành phần nào khác.

Hình 2-5 Thiết bị I /O Board Nexys 2

Các ngõ vào: Slide Switches và Pushbuttons

Bốn nút nhấn và 8 công tắc được cung cấp cho ngõ vào. Các nút nhấn thường ở mức thấp, khi được nhấn sẽ lên mức cao. Công tắc tạo ra những ngõ vào mức cao hay mức thấp phụ thuộc vào vị trí của chúng. Các nút nhấn và công tắc ngõ vào được sử dụng các điện trở nối tiếp để bảo vệ tránh hiện tượng đoản mạch (hiện tượng đoản mạch xảy ra khi chân của FPGA được gán một nút nhấn hoặc công tắc vô tình được định nghĩa như một ngõ ra.



Hình 2-6 Mạch và thiết bị I / O của Board Nexys 2

Ngõ ra: LEDs

Tám LED đơn được cung cấp cho ngõ ra của mạch. Cực dương LED được điều khiển từ FPGA thông qua điện trở 390-ohm, do đó, đầu ra mức logic '1 ' sẽ sáng với dòng khoảng 3-4mA. LED thứ chín là được cung cấp như một LED chỉ thị nguồn, và LED 10 cho biết trạng thái chương trình FPGA.

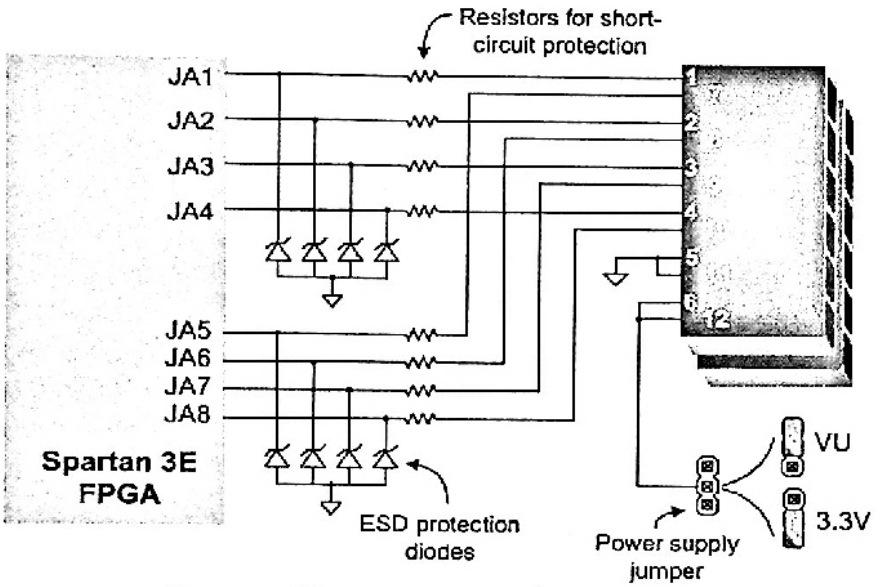


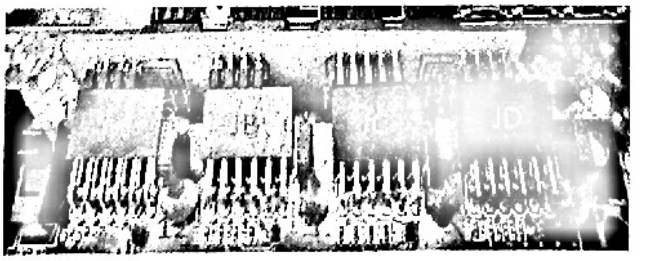
Figure 23: Nexys2 Pmod connector circuits

Hình 2-7 sơ đồ nguyên lý của các cổng I/O mở rộng của kit

Table 3: Nexys2 Pmod Connector Pin Assignments							
Pmod JA		Pmod JB		Pmod JC		Pmod JD	
JA1: L15	JA7: K13	JB1: M13	JB7: P17	JC1: G15	JC7: H15	JD1: J13	JD7: K14[1]
JA2: K12	JA8: L16	JB2: R18	JB8: R16	JC2: J16	JC8: F14	JD2: M18	JD8: K15[2]
JA3: L17	JA9: M14	JB3: R15	JB9: T18	JC3: G13	JC9: G16	JD3: N18	JD9: J15[3]
JA4: M15	JA10: M16	JB4: T17	JB10: U18	JC4: H16	JC10: J12	JD4:; P18	JD10: J14[4]

Notes: [1] shared with LD3 [2] shared with LD3 [3] shared with LD3 [4] shared with LD3

Bảng 2-2 Địa chỉ các chân I/O mở rộng

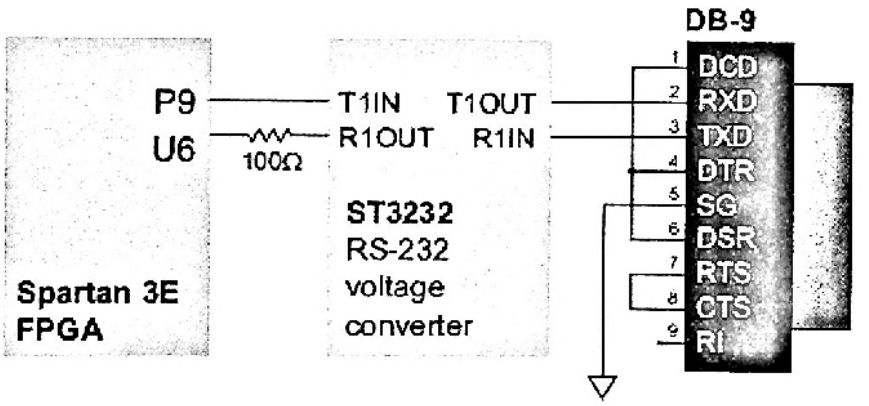


Hình 2-8 cổng I/O thực tế

2.5 Cổng nối tiếp

Nexys 2 chưa một cổng nối tiếp cơ bản 2 dây chuyển đổi điện áp trên một vi điện tử ST3232. Vi điện tử ST3232 chuyển đổi các mức tín hiệu được sử dụng bởi giao tiếp RS-232 (từ -12V – 3V cho mức logic '1' và từ 3V – 12V cho mức logic '0') tới tín hiệu 3.3V dùng trong FPGA. Do chỉ có hai tín hiệu được kết nối (RXD và TXD), cổng nối tiếp trên FPGA điều khiển chỉ có thể sử dụng phần mềm giao thức bắt tay (XON, XOFF). Cổng nối tiếp Nexys 2 hữu ích trong nhiều ứng dụng, và đặc biệt là dùng để sửa lỗi và làm việc với vi xử lý nhúng MicroBlaze của Xilinx.

Hai thiết bị được kết nối với hai đầu của một sợi cáp nối tiếp được gọi là thiết bị đầu cuối dữ liệu (DTE) và thiết bị truyền thông dữ liệu (DCE). DCE đầu tiên được hiểu là một Modem, nhưng bây giờ nhiều thiết bị được kết nối với máy tính như một DCE. Một thiết bị DTE "nguồn" sử dụng một kết nối DB-9 đầu đực và một thiết bị DCE "ngoại vi" sử dụng một kết nối DB-9 đầu cái. Hai thiết bị DTE có thể được kết nối thông qua một cáp nối tiếp chỉ khi hai và ba dòng (RXD và TXD)chéo nhau, sản xuất những gì được biết đến như một cáp modem không. Một thiết bị DTE và DCE có thể được kết nối với một cáp thẳng. Nexys2 được cấu hình như một thiết bị DCE, với giả thiết đặc trưng nhất là nó sẽ được kết nối với một thiết bị DTE như một máy tính.



Hình 2-9 Mạch cổng nối tiếp Nexys 2

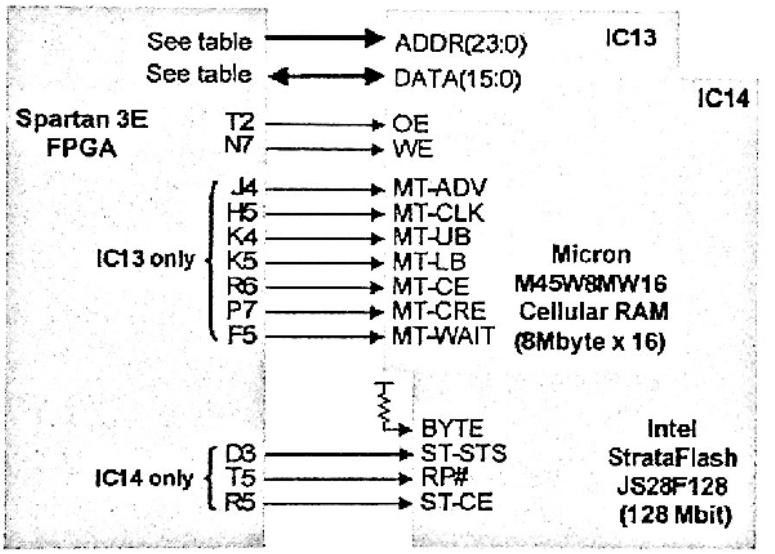
2.6 Bộ nhớ

Board Nexys 2 có thiết bị RAM và ROM ngoài. RAM ngoài là một Micro 128Mbit M45W8MW16 Cellular RAM giả tĩnh DRAM thiết bị được thiết lập 8Mbytes x 16bits. Nó có thể hoạt động như là một SRAM không đồng bộ điển hình đọc và viết chu kỳ 70ns, hoặc như một bộ nhớ đồng bộ với một đường dẫn 80MHz. Khi hoạt động như là một SRAM không đồng bộ, RAM Cellular tự động làm tươi mảng DRAM nội của nó, cho phép một thiết kế điều khiển đơn giản hóa bộ nhớ (tương tự như bất kỳ SRAM) trong FPGA. Khi hoạt động ở chế độ đồng bộ, truyền liên tục lên đến 80MHz là có thể.

ROM ngoại là một Intel 128Mbit TE28F128J3D74-110 StrataFlash thiết bị được thiết lập 8Mbytes 16bits x. Bên trong, nó có chứa 128 khối có thể từng khối một bị xóa, và nó hỗ trợ chu kỳ đọc 110ns, với 25ns đọc chế độ trang trong khối. Nó có một bộ đệm 32-byte viết nội bộ có thể được viết với thời gian chu kỳ đánh giá 70ns, và bộ đệm 32-byte có thể được chuyển giao cho các mảng Flash trong 218us (điển hình).

Cả hai thiết bị đều sử dụng chung 16-bit bus dữ liệu và 24-bit bus địa chỉ. RAM Cellular là byte định địa chỉ bằng cách sử dụng các tín hiệu byte cao và byte thấp (MT-UB và MT-LB), nhưng StrataFlash được cấu hình cho 16 byte hoạt động (nó không phải là byte địa chỉ). Tín hiệu cho phép xuất ra(OE) và ghi (WE) tín được sử dụng bởi cả hai thiết bị, nhưng mỗi thiết bị có con chip cho phép tín hiệu (CE) riêng. Ngoài ra, bộ nhớ RAM Cellular có clock (MT-CLK), đợi (MT-WAIT), địa chỉ hợp lệ (MT-ADV) và cho phép thanh ghi điều khiển (MT_CRE) tín hiệu có sẵn cho FPGA để sử dụng với truyền đồng bộ, và StrataFlash có thiết lập lại (RP #) tình trạng và (STS) tín hiệu truyền đến FPGA.

Mã nguồn VHDL sẵn có trong một thiết kế tham khảo được đăng trên trang web Digilent để minh họa cho việc sử dụng các thiết bị này. Một hệ thống tập tin cơ sở xây dựng cũng có sẵn để sử dụng các thiết bị này với công cụ EDK Xilinx và bộ xử lý lõi MicroBlaze, cả hai đều có sẵn từ Xilinx.Toàn bộ thông tin có sẵn cho cả hai thiết bị từ các trang web nhà sản xuất.



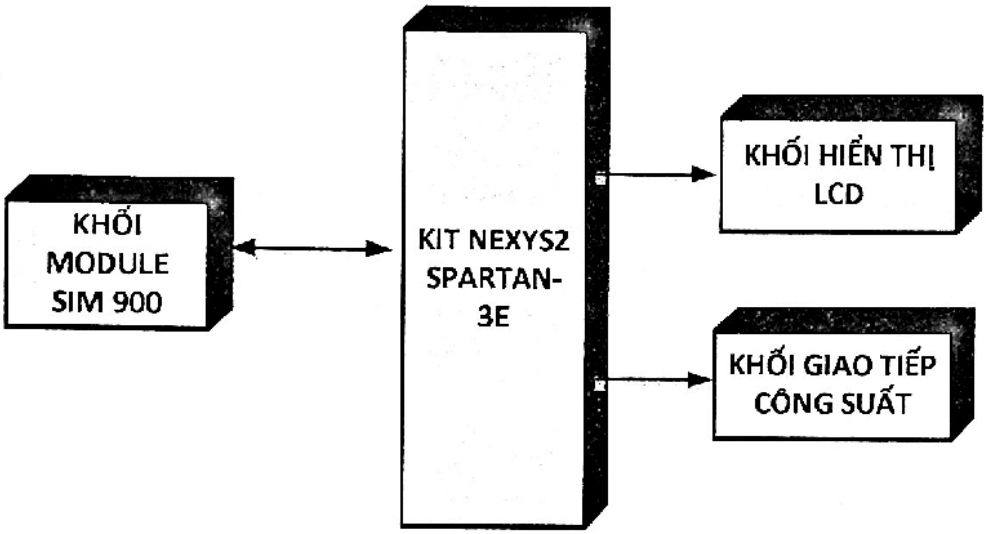
Hình 2-10 Mạch bộ nhớ Nexys 2

Address signals			Data signals	
ADDR0: NA	ADDR8: H6	ADDR16: M5	DATA0: L1	DATA8: L3
ADDR1: J1	ADDR9: F1	ADDR17: E2	DATA1: L4	DATA9: L5
ADDR2: J2	ADDR10: G3	ADDR18: C2	DATA2: L6	DATA10: M3
ADDR3: H4	ADDR11: G6	ADDR19: C1	DATA3: M4	DATA11: M6
ADDR4: H1	ADDR12: G5	ADDR20: D2	DATA4: N5	DATA12: L2
ADDR5: H2	ADDR13: G4	ADDR21: K3	DATA5: P1	DATA13: N4
ADDR6: J5	ADDR14: F2	ADDR22: D1	DATA6: P2	DATA14: R3
ADDR7: H3	ADDR15: E1	ADDR23: K6	DATA7: R2	DATA15: T1

Bảng 2-3 Gán chân Bus dữ liệu và địa chỉ bộ nhớ

THIẾT KẾ PHẦN CỨNG

3.1 SƠ ĐỒ KHỐI



Hình 3-1 Sơ đồ khối của hệ thống

Chức năng từng khối:

- Khối Module SIM 900 có chức năng thu phát tin nhắn.
- Khối điều khiển dùng FPGA, Spartan-3E, giao tiếp với khối GSM để nhận tin nhắn và gởi tin nhắn. Sau khi nhận tin nhắn sẽ tiến hành giải mã điều khiển thiết bị theo yêu cầu, gởi tín hiệu trả lời sau khi điều khiển xong.
- Khối hiển thị dùng để hiển thị các thông tin điều khiển, tin nhắn gởi đến.
- Khối giao tiếp công suất để điều khiển thiết bị.

3.2 CHI TIẾT TỪNG KHỐI NGOẠI VI

3.2.1 Module sim 900

Giới thiệu về module sim 900: là một modem GSM là một modem wireless, nó làm việc cùng với một mạng wireless GSM. Một modem wireless thì cũng hoạt động giống như một modem quay số. Điểm khác nhau chính ở đây là modem quay số thì truyền và nhận dữ liệu thông qua một đường dây điện thoại cố định trong khi đó một modem wireless thì việc gửi nhận dữ liệu thông qua sóng.

Giống như một điện thoại di động GSM, một modem GSM yêu cầu 1 thẻ sim với một mạng wireless để hoạt động.

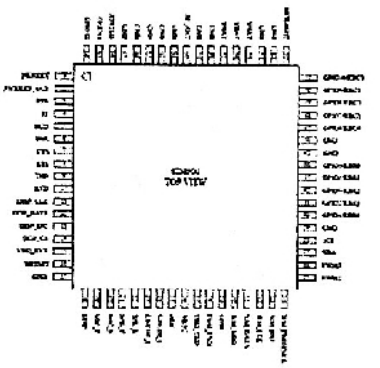
Module Sim900 là một trong những loại modem GSM. Nhưng Module Sim900 đựoc nâng cao hơn có tốc độ truyền dữ liệu nhanh hơn. Nó sử dụng công nghệ GSM/GPRS hoạt động ở băng tầng EGSM 900Mhz, DCS 1800 Mhz và PCS 1900Mhz, tính năng GPRS của Module Sim900 có nhiều lớp:

- 8 lớp điện dung
- 10 lớp điện dung

Và hỗ trợ GPRS theo dạng đồ thị mã hóa CS-1, CS-2, CS-3 và CS-4

Nguyên lý hoạt động: Module sim 900 có chức năng như chiếc điện thoại di động, nó nhận tin nhắn thông qua mạng GSM, sau đó lưu tin nhắn vào sim rồi chuyển xuống kit Nexys2 thông qua chân TX. Đồng thời module sim cũng nhận dữ liệu từ kit qua chân RX, gửi tin nhắn trả lời đến thuê bao thông qua mạng GSM.

3.2.1.1 Đặc điểm của module sim 900



Hình 3-2 Module sim 900

1. Tính năng của Module Sim900:
2. Nguồn cung cấp : 3.4 – 4.5 VDC
3. Điện năng tiêu thụ trong chế độ "ngủ": 1.5 mA
4. Tương thích với GSM phase 2/2+
5. Tự động tìm băng tần phù hợp trong 4 băng tần GSM 850MHz, EGSM 900MHz, DCS 1800MHz và PCS 1900MHz
6. Lớp GSM: Small MS
7. Nhiệt độ hoạt động: -30°C đến +80°C
8. Tốc độ GPRS:

- Downloads data: 84.6 kpbs
- Uploads data: 42.8 kpbs

9. SMS:

- MT, MO, CB, Text and PDU mode
- Bộ nhớ SMS: Sim card

10. FAX:

- Nhóm 3 loại 1

11. Sim card:

- Hỗ trợ sim card: 1,8v ; 3v

12. Anten ngoài:

- Kết nối thông qua anten ngoài 500km hoặc đế anten

13. Âm thanh:

- Dạng mã hòa âm thanh.
- Mức chế độ (ETS 06.20)
- Toàn bộ chế độ (ETS 06.10)
- Toàn bộ chế độ tăng cường (ETS 06.50/ 06.06/ 06.80)
- Loại bỏ tiếng dội

14. Giao tiếp nối tiếp và sự ghép nối:

- Cổng nối tiếp: 7 Cổng nối tiếp(ghép nối)
- Cổng kết nối có thể Sd với CSD Fax, GPRS và gửi lệnh ATCommand tới mudule điều khiển
- Cổng nối tiếp có thể Sd chức năng giao tiếp
- Hỗ trợ tốc độ truyền 1200 BPS tới 115200 BPS
- Cổng hiệu chỉnh lỗi: 2 cổng nối tiếp TXD và RXD
- Cổng hiệu chỉnh lỗi chỉ sử dụng sữa lỗi

14. Quản lý danh sách:

- Hỗ trợ mẫu danh sách: SM, FD, LD, RC,ON, MC

16. Sim Application toolkit:

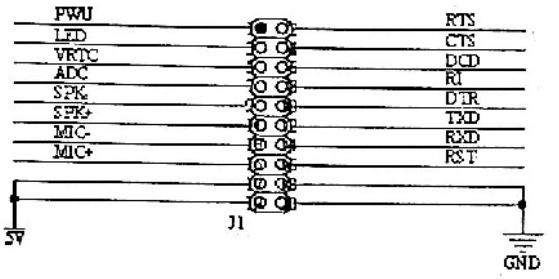
- Hỗ trợ SAT loại GSM 11,14 bản 99

17. Đồng hồ thời gian thực:

- Người cài đặt

20. Chương trình hoàn thiện.

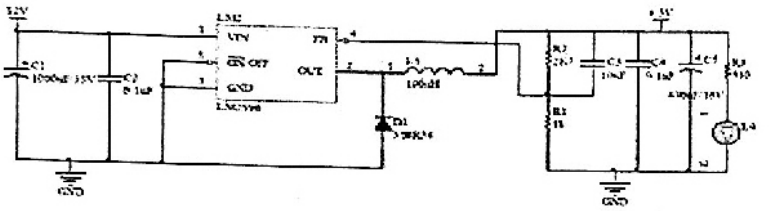
3.2.1.2 Sơ đồ các chân ngõ ra của module sim 900



Hình 3-3 Sơ đồ chân của Module sim 900

- ✓ Chân 7 (ADC) : Chân vào của bộ chuyển đổi tín hiệu tương tự sang số
- ✓ Chân 5 (VRTC): Current input for RTC when the battery is not supplied for the system.Current output for backup battery when the main battery is present and the backup battery is in low voltage state.
- ✓ Chân 1 (PWU): chân này dùng để điều khiển hệ thống bật/tắt
- ✓ Chân 17, 19 (VCC) : nguồn cung cấp cho thẻ sim.
- ✓ Chân 16 (RST) :chân reset cho mạch sim
- ✓ Chân 10 (DTR) : chân đầu cuối dữ liệu
- ✓ Chân 2 (RTS) : Request to send
- ✓ Chân 4 (CTS) : Clear to send
- ✓ Chân 8 (RI) : Ring indicator
- ✓ Chân 14 (RXD) : đầu ra dùng để điều chỉnh trong nhận dữ liệu
- ✓ Chân 12 (TXD) : đầu ra dùng để điều chỉnh trong truyền dữ liệu
- ✓ Chân 18, 20 (GND) : chân max
- ✓ Chân 9 (SPK-), 11 (SPK+) : chân output
- ✓ Chân 13 (MIC-), 15 (MIC+) : chân input

Mạch nguồn cung cấp cho Module Sim900:



Hình 3-4 Mạch nguồn cung cấp cho module sim 900

3.2.1.3 At command

Một số tập lệnh AT thường dùng:

- Lệnh AT<cr> : kiểm tra đường truyền
- Lệnh AT+CMGF=[<mode>] <cr> : lệnh cấu hình tin nhắn

  <mode> : 0 dạng dữ liệu PDU

  1 dạng dữ liệu kiểu text
- Lệnh AT&W[<n>] : lệnh lưu cấu hình cho sim900
- Lệnh ATD[số điện thoại]<cr> : gọi điện thoại.
- Lệnh AT+CMGD = <index> <cr> : lệnh xóa tin nhắn

  <index>: vị trí ngăn nhớ lưu tin nhắn
- Lệnh AT+CMGR=<index><cr> : lệnh đọc tin nhắn

  <index> : số nguyên, đó là vị trí ngăn nhớ chứa tin nhăn cần đọc
- Lệnh AT+CMGS="số điện thoai": lệnh gửi tin nhắn
- Lệnh bắt đầu kết nối TCP hay UDP

  AT+CIPSTART=<mode>,[<IPaddress>,<domain name>],<port>

  Trong đó:

  <mode> : tham số báo kiểu kết nối TCP hay UDP

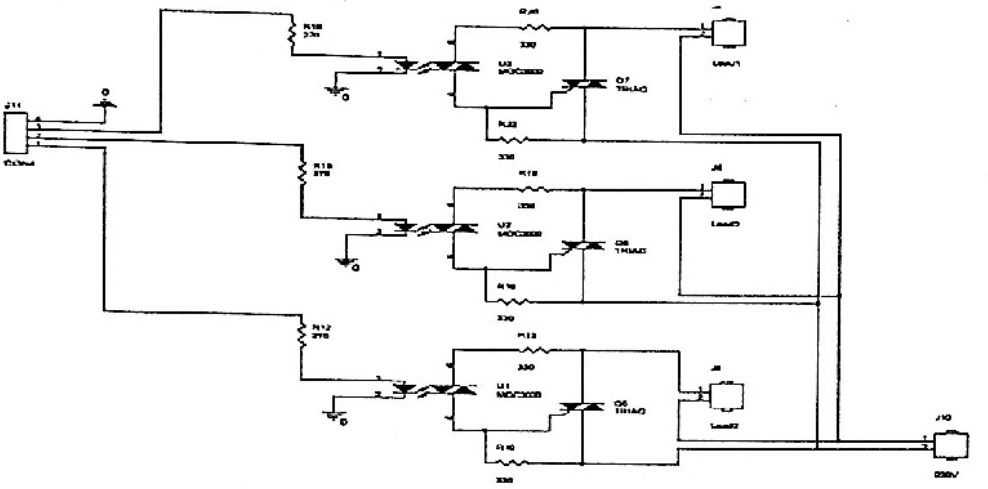
  <IP address> : địa chỉ IP của người điều khiển từ xa

  <domain name> : tên miền của người điều khiển từ xa

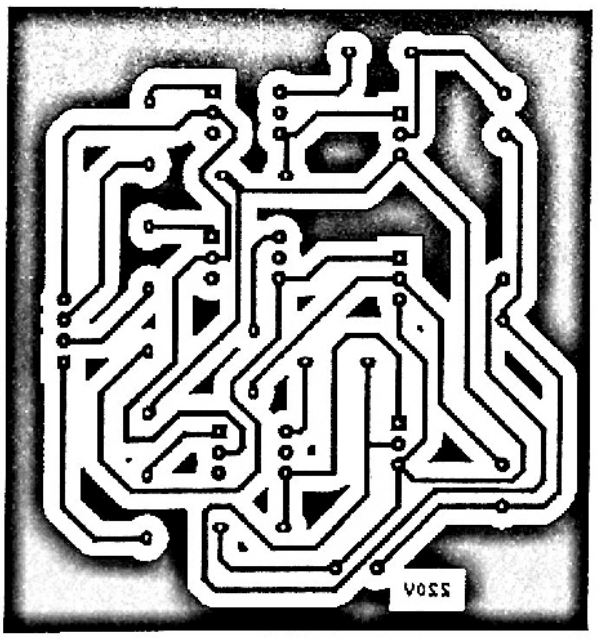
3.2.2 Khối công suất

Giao tiếp để điều khiển 3 thiết bị là 3 bóng đèn 220V.

Sơ đồ mạch nguyên lý:



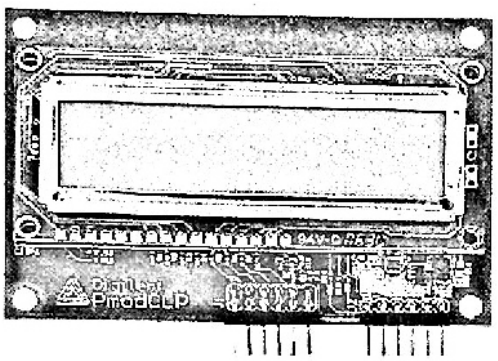
Hình 3-5 Mạch nguyên lý của mạch công suất



Hình 3-6 Sơ đồ mạch in của mạch công suất

3.2.3 Khối hiển thị LCD

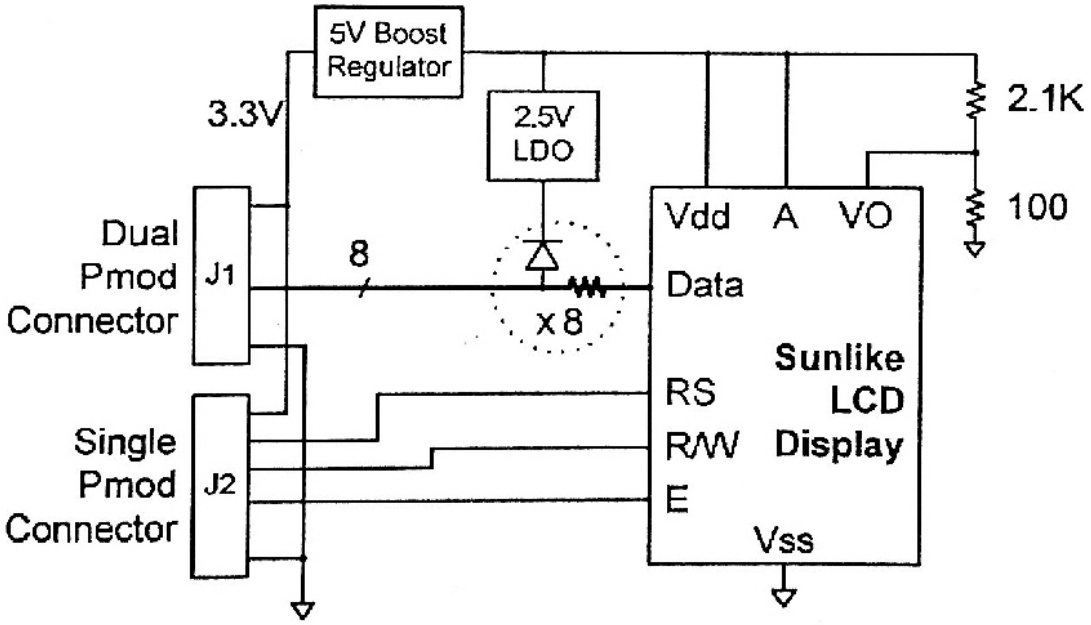
Module LCD (pmodCLP) là LCD 16x2 ký tự sử dụng 2 bộ kết nối Pmod sử dụng nguồn 3.3V, giao tiếp dữ liệu 8 bit song – song với các bo hệ thống. Module LCD sử dụng LCD của Sunlike sử dụng bộ điều khiển Samsung KS0066 hoặc bộ điền khiển tương đương. Module LCD này có thể kết nối với bất kỳ hệ thống nào của hang Digilent để hiển thị các ký tự trên LCD.



Hình 3-7 Module LCD 16x2

Module PmodCLP có thể sự dụng nguồn 3.3V hoặc 5V. Do LCD của Sunlike sử dụng nguồn 5V nên trong module Pmode CLP có mạch chuyển đổi nguồn từ 3.3V lên 5V có ổn áp để cấp cho LCD nếu dùng nguồn 3.3V. Mạch điện chuyển đổi điện áp dùng để tương thích các tín hiệu dữ liệu với điện áp 3.3V của các bo hệ thống bất chấp nguồn cung cấp là 3.3V hoặc 5V.

Sơ đồ khối của module PmodCLP như hình



Hình 3-8 Module PmodCLP.

Module PmodCLP kết nối với các bo hệ thống qua một bộ kết nối một hang chân và một bộ kết nối 2 hàng chân bao gồm 8 đường dữ liệu và 3 đường tín hiệu điều khiển. Nguồn cung cấp cho LCD từ 3.3V đến 5V qua mạch khuếch đại và ổn áp tạo ra điện áp 5V.

Table 1. PmodCLP Connector Signals		
Pin No.	Symbol	Signal Description
J1.5, J1.11, J2.5	Vss	Signal ground
J1.6, J1.12, J2.6	Vdd	Power supply (5V)
J2.1	RS	Register select: high for data transfer, low for instruction register
J2.2	R/W	Read/write signal: high for read mode, low for write mode
J2.3	E	Read/write strobe: high for read OE; falling edge writes data
J2.4	No Connect	Optional back-light enable (not connected on PmodCLP)
J1.1,2, 3, 4 J1.7,8,9,10	Data Bus	Bi-directional 8-bit data bus

Bảng 3-1 Liệt kê các chân giao tiếp với các bộ kết nối

THIẾT LẬP CẤU HÌNH CHO FPGA VÀ LƯU ĐỒ CHƯƠNG TRÌNH ĐIỀU KHIỂN

4.1 GIỚI THIỆU PHẦN MỀM EDK CỦA HÃNG XILINX

Khi sử dụng FPGA của Xilinx theo hướng ứng dụng nhúng sử dụng MicroBlaze hay PowerPC, chúng ta dùng phần mềm EDK của gói phần mềm ISE của Xilinx.

EDK là một bộ công cụ phần mềm có chứa hàng chục lõi IP đủ để xây dựng nên một hệ thống nhúng dựa trên vi điều khiển mềm MicroBlaze hoặc lõi cứng PowerPC (dòng Virtex). Các lõi IP có thể được xây dựng trên ngôn ngữ VHDL hay Verilog. EDK xây dựng một hệ thống phần cứng hoàn chỉnh dựa trên vi điều khiển mà sau đó tất cả sẽ được đặt vào trong một FPGA (đương nhiên là của Xilinx). Trong hệ thống này bao gồm CPU (MicroBlaze hoặc PowerPC), các ngoại vi như UART, SPI, Interrupt, Timer và nhiều ngoại vi khác (bao gồm cả PCI). Sau khi xây dựng hệ thống phần cứng hoàn chỉnh (lúc này vẫn chỉ ở dạng sơ đồ trên PC), chúng ta sẽ tiếp tục phát triển chương trình cho hệ thống. Chương trình có thể viết bằng C/C++ một cách nhanh chóng. Tất cả phần cứng, phần mềm sẽ được công cụ EDK (có hỗ trợ của ISE) chuyển thành file *.bit cho phép cấu hình cho FPGA. Lúc này thì FPGA đã trở thành một hệ thống đầy đủ. Nếu thêm phần analog thì ta phải thiết kế thêm bên ngoài, EDK có hỗ trợ IP cho ADC/DAC nhưng tốc độ không cao.

Bộ phần mềm EDK gồm có chương trình XPS (Xilinx Platform Studio) và SDK (Software Development Kit).

Để tạo một hệ thống nhúng bằng phần mềm EDK, ta thông qua các quá trình:

- Tạo BSB (Blank System Builder) sử dụng XPS: Đây là giai đoạn xây dựng hệ thống phần cứng và các ngoại vi cho dự án dựa trên một CPU được tích hợp sẵn (CPU này có thể là Microblaze hoặc PowerPC).
- Viết chương trình nhúng cho các ngoại vi: Chương trình có thể sử dụng phần mềm SDK (trong bộ EDK) hoặc thực hiện viết trực tiếp trên XPS.

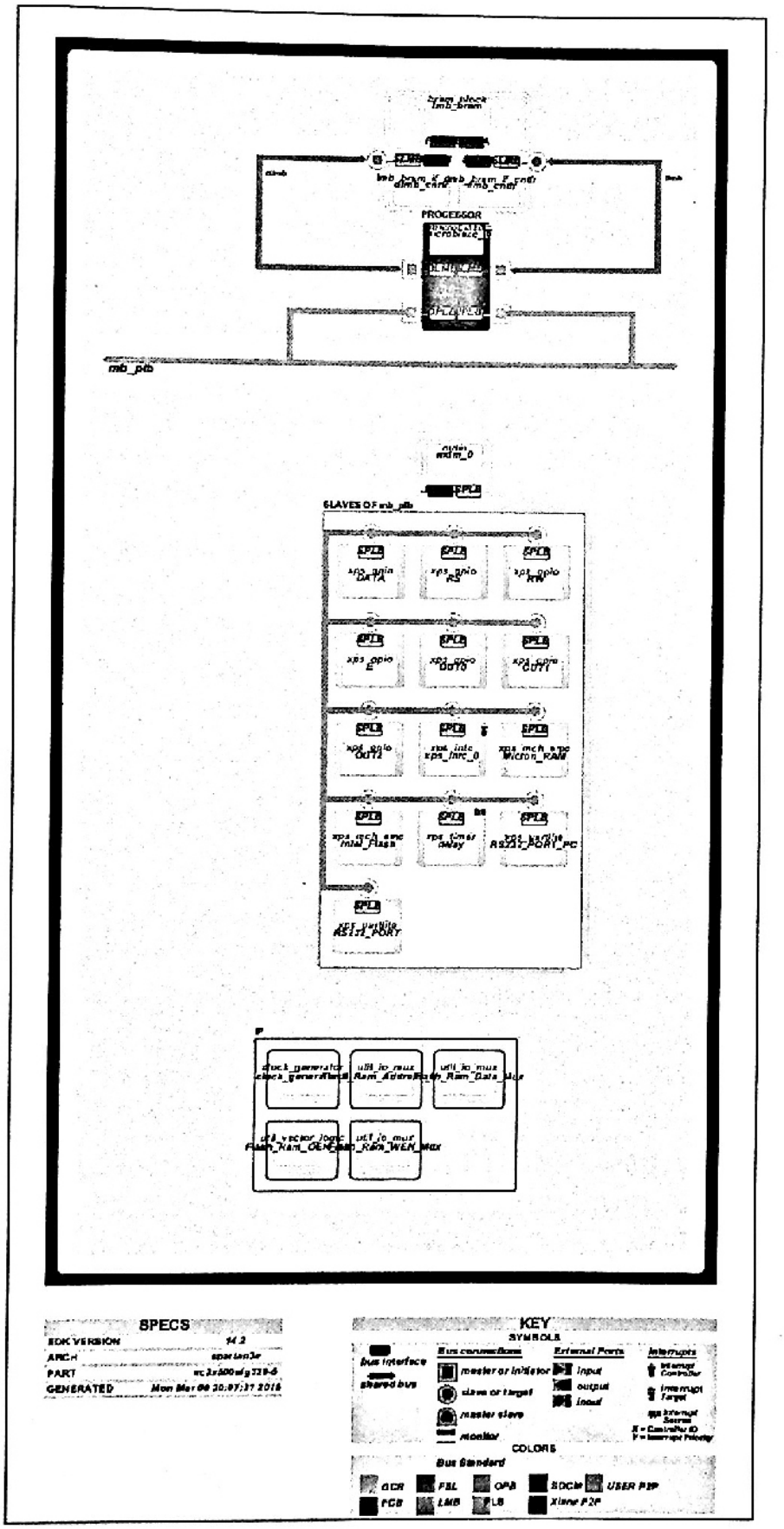
4.2 THIẾT KẾ CẤU HÌNH TRÊN FPGA DÙNG XPS

4.2.1 Yêu cầu

Thiết kế được phần cứng của hệ thống điều khiển thiết bị bằng tin nhắn trên kit **Nexys™2 Spartan-3E FPGA Board** gồm:

- Để giao tiếp với module sim 900 cần một cổng uart, một timer để định thời và một interrupt.
- Để giao tiếp với LCD cần 1 bus data (8bit) và 3 đường điều khiển là RS, RW, E.
- Để giao tiếp với thiết bị điều khiển (3 thiết bị): cần 3 tín hiệu I/O.

4.2.2 Sơ đồ phần cứng yêu cầu



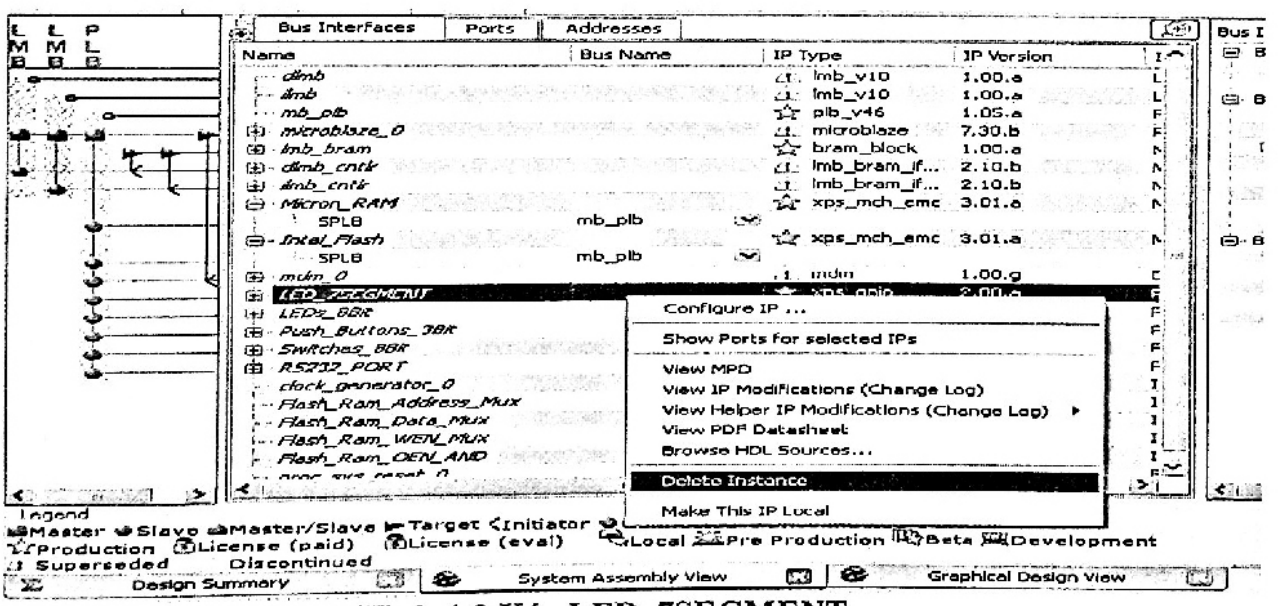
Hình 4-1 Sơ đồ phần cứng

4.2.3 Các bước thực hiện:

- **Bước 1: Tải cấu hình IP của kit Nexys™2 Spartan-3E FPGA Board đã được Xilink thiết kế sẵn:**
  - ✓ Truy cập vào trang web của Xilink có miền địa chỉ như sau: http://www.digilentinc.com/Products/Detail.cfm?NavPath=2,400,789&Prod=NEXYS2.
  - ✓ Download cấu hình IP của của kit Nexys™2 Spartan-3E FPGA Board như hình sau:
    - Chọn " EDK project that demonstrates how to connect the BSB shared buses through I/O multiplexers, allowing access to both RAM and FLASH memories on the Nexys2-500. Made using EDK 11 " kích Download để tải thư mục cấu hình về.

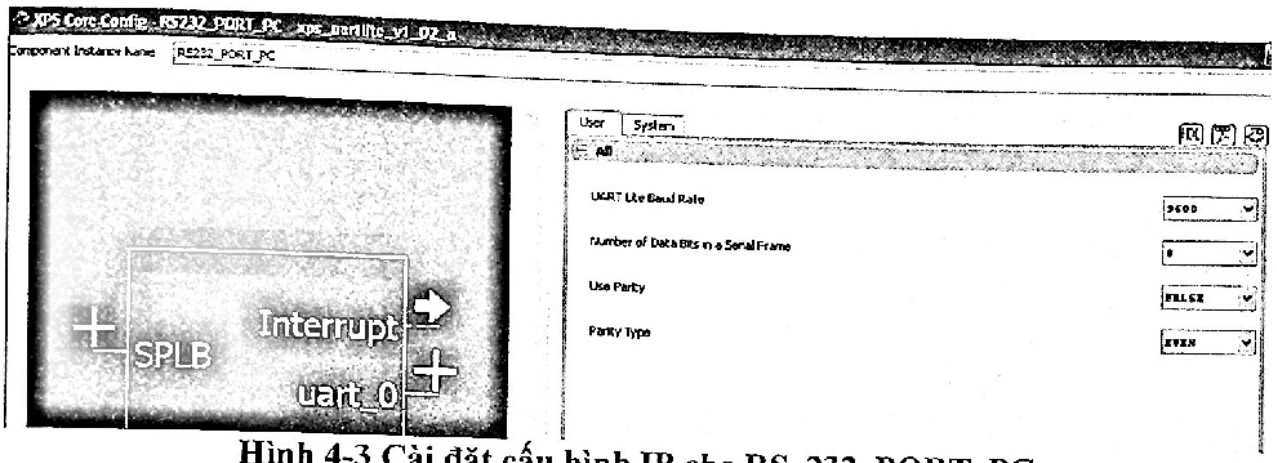
- **Bước 2: Khởi động chương trình XPS từ thư mục thiết kế IP Nexys-2 đã download trên trang Xilinkx :** cập nhập nâng cấp lõi vi điều khiển microblaze và IP nếu dùng phần mềm EDK có phiên bản cao hơn 11.

- **Bước 3 : Thêm IP vào hệ thống.**
  - ✓ Xoá các IP không dùng tới.
    - Xoá **LED_7SEGMENT**
      - + Nhấp phải chuột vào **LED_7SEGMENT**
      - + Chọn **Delete Instance**



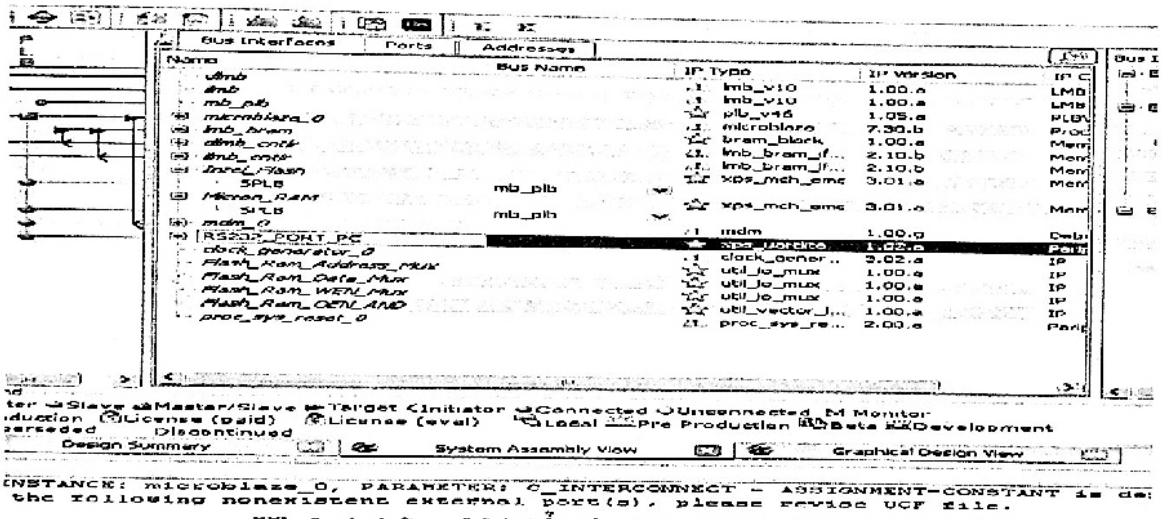
Hình 4-2 Xóa LED_7SEGMENT.

- ✓ Tương tự xóa các IP **LEDs_8Bit , Push_Buttons_3Bit , Switches_8Bit** giống như xóa **LED_7SEGMENT .**
- ✓ *Đổi tên RS232_PORT thành RS232_PORT_PC:*
  - Nhấp đúp chuột trái vào RS232_PORT đổi tên thành RS232_PORT_PC và định cấu hình IP như hình dưới :



Hình 4-3 Cài đặt cấu hình IP cho RS_232_PORT_PC

- Ta được IP mới như hình sau:

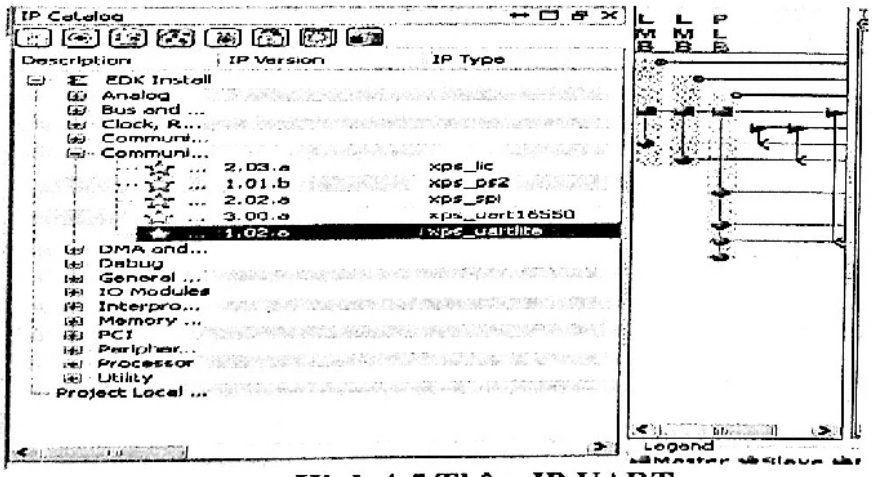


Hình 4-4 Sau khi đổi tên thành RS_232_PORT_PC

✓ Thêm IP vào hệ thống

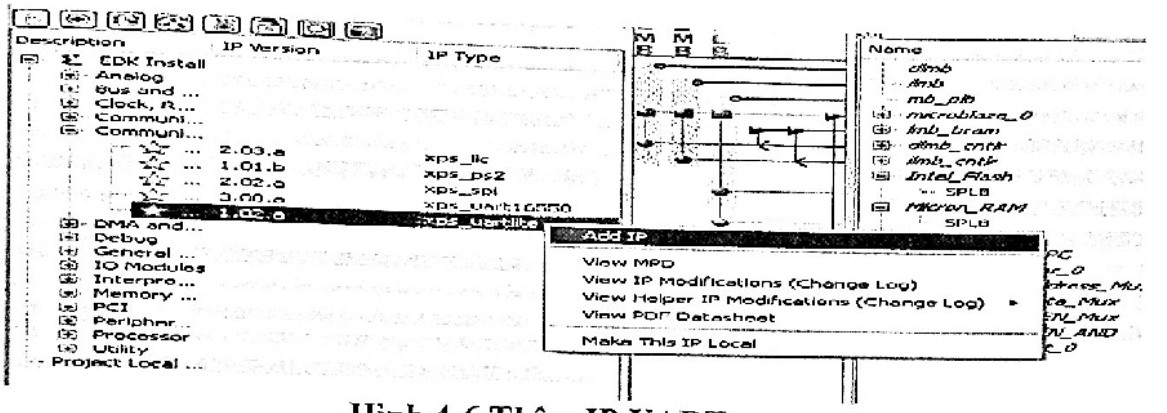
Thêm RS232_PORT

- Chọn thẻ **IP catalog .**
- Mở rộng **Communication Low-Speed**



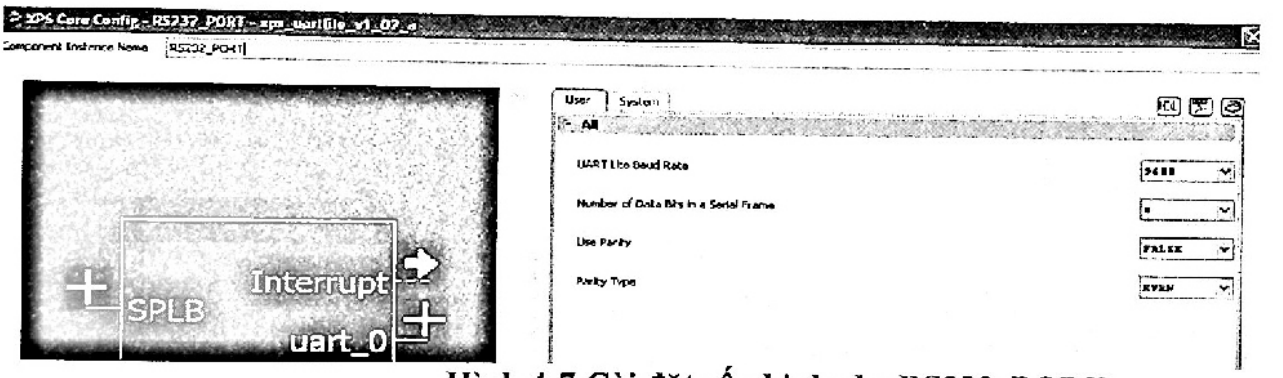
Hình 4-5 Thêm IP UART.

- Nhấp phải chuột vào **xps_uartlite.**
- Chọn **Add IP.**



Hình 4-6 Thêm IP UART.

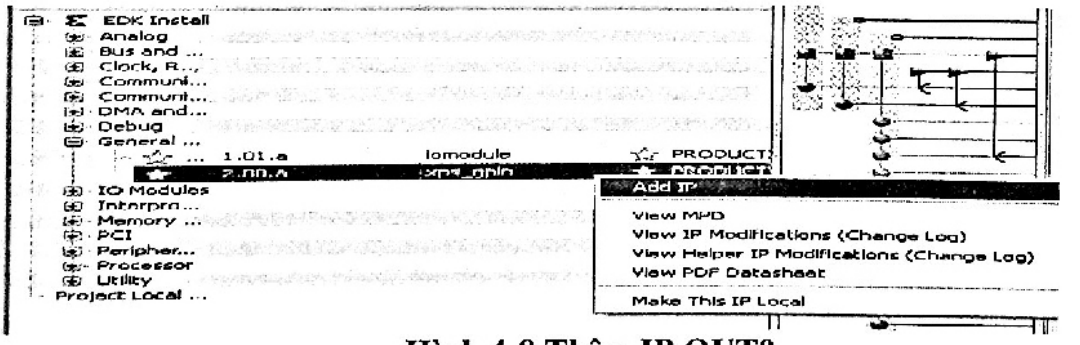
- Sửa tên thành **RS232_PORT** trong ô **Component Instance Name.** Định cấu hình cho IP như hình dưới:



Hình 4-7 Cài đặt cấu hình cho RS232_PORT

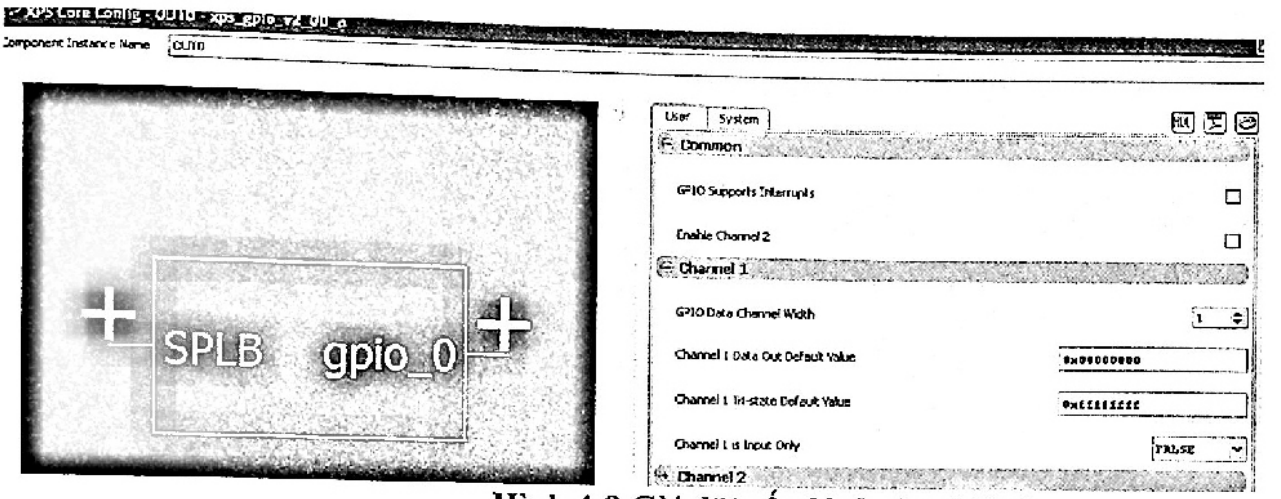
Thêm GPIO điều khiển thiết bị thứ 0

- Mở rộng General Purpose IO
- Nhấp phải chuột vào **xps_gpio.**
- Chọn **Add IP.**



Hình 4-8 Thêm IP OUT0

- Sửa tên thành OUT0 trong ô **Component Instance Name.** Định cấu hình cho IP như hình dưới:



Hình 4-9 Cài đặt cấu hình cho OUT0

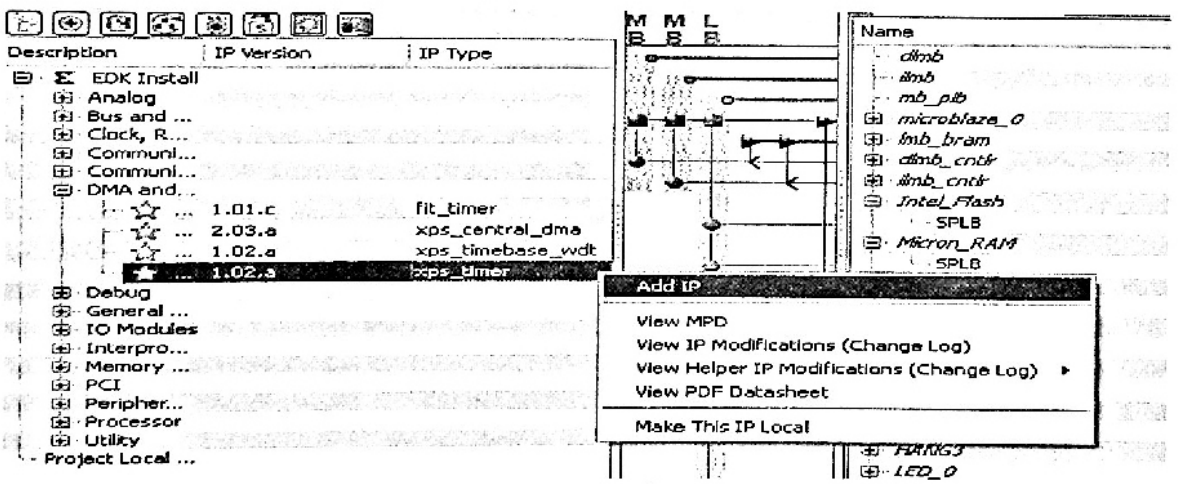
Tương tự như các bước trên và cài đặt cấu hình cho các IP OUT0, OUT1, OUT2, RS, RW, E, DATA.

Thêm timer delay

- Mở rộng **DMA and Timer**

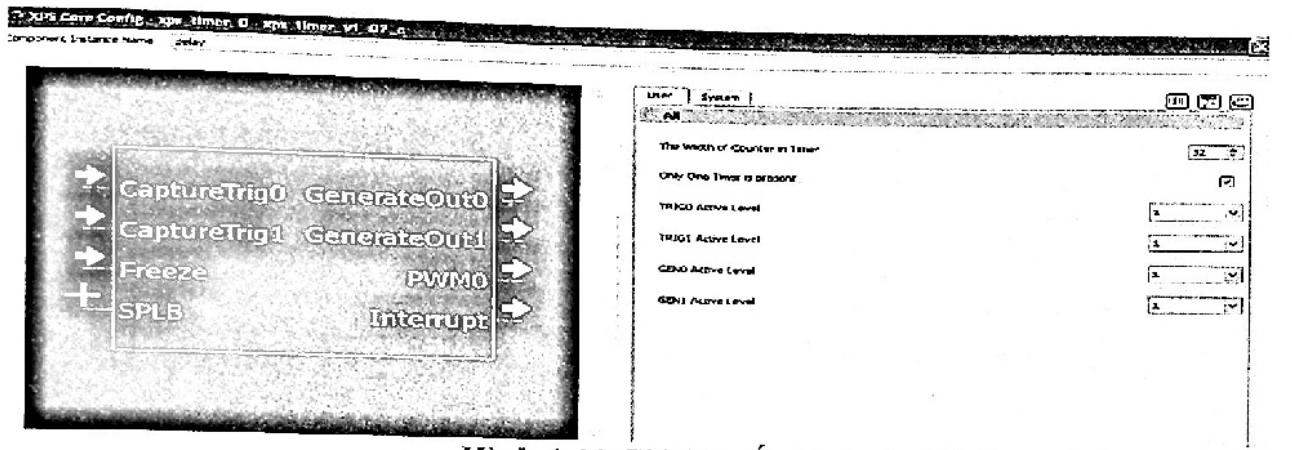
✓ Nhấp phải chuột vào **xps_timer.**

✓ Chọn **Add IP.**



Hình 4-10 Thêm IP timer delay

- Sửa tên thành **delay** trong ô **Component Instance Name**. Định cấu hình cho IP như hình dưới:



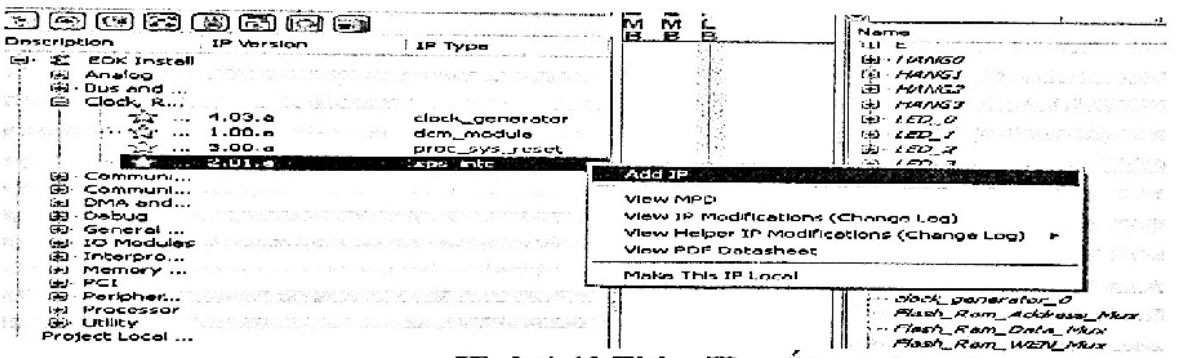
Hình 4-11 Cài đặt cấu hình cho IP timer delay

Thêm IP ngắt

- Mở rộng Clock,Reset and Interupt

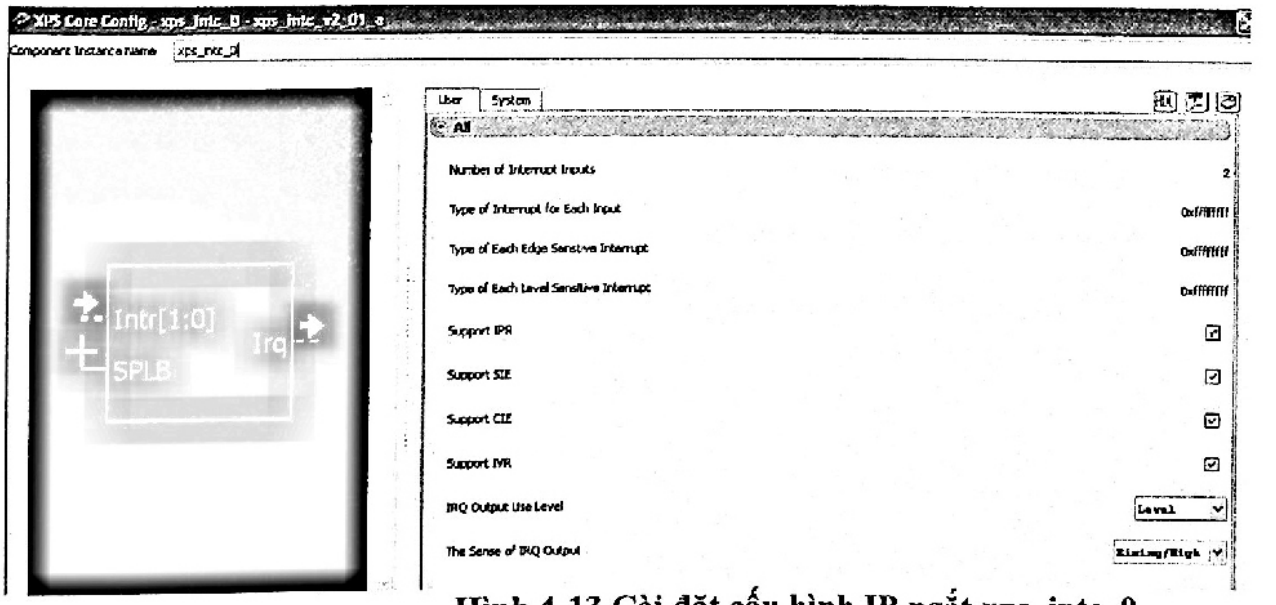
✓ Nhấp phải chuột vào **xps_intc.**

✓ Chọn **Add IP.**



Hình 4-12 Thêm IP ngắt xps_intc_0

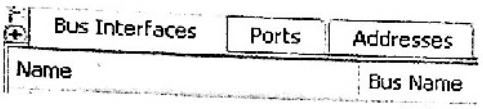
- Định cấu hình cho IP như hình dưới:



Hình 4-13 Cài đặt cấu hình IP ngắt xps_intc_0

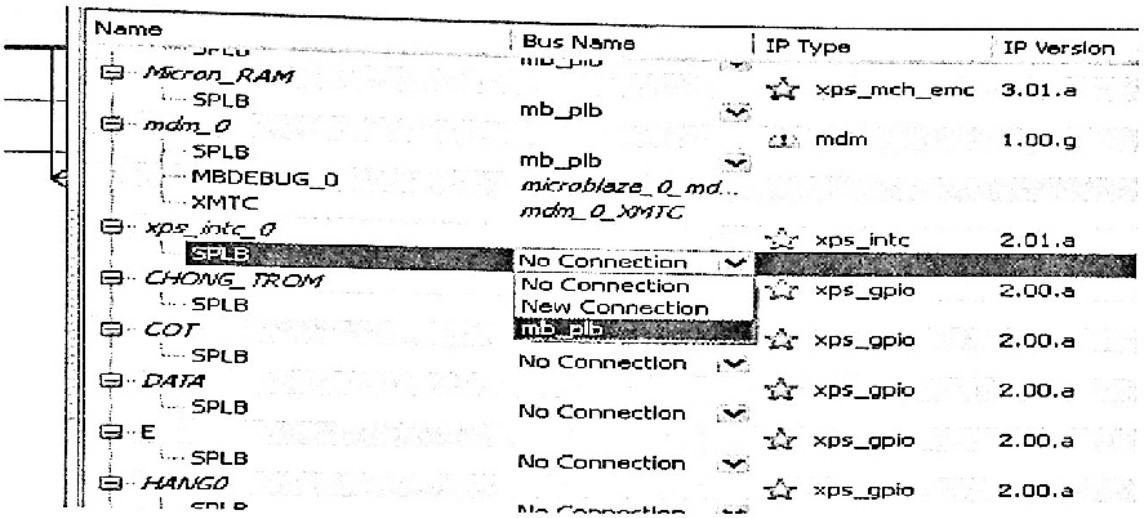
- **<u>Bước 4</u>: Kết nối IP vào hệ thống bus.**

✓ Chọn thẻ **Bus Interfaces .**

✓ Chọn dấu '+' tại 

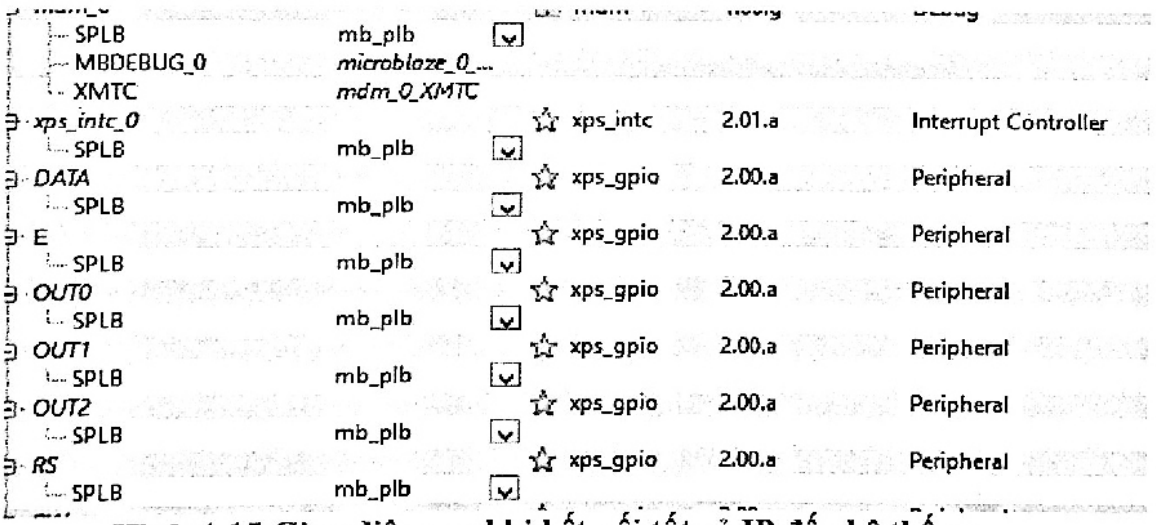
để mở rộng tất cả IP có trên hệ thống.

✓ Xps_intc_0 : **SPLB** chọn **mb_plb** để kết nối IP vào hệ thống



Hình 4-14 Chọn chế độ kết nối của SPLB

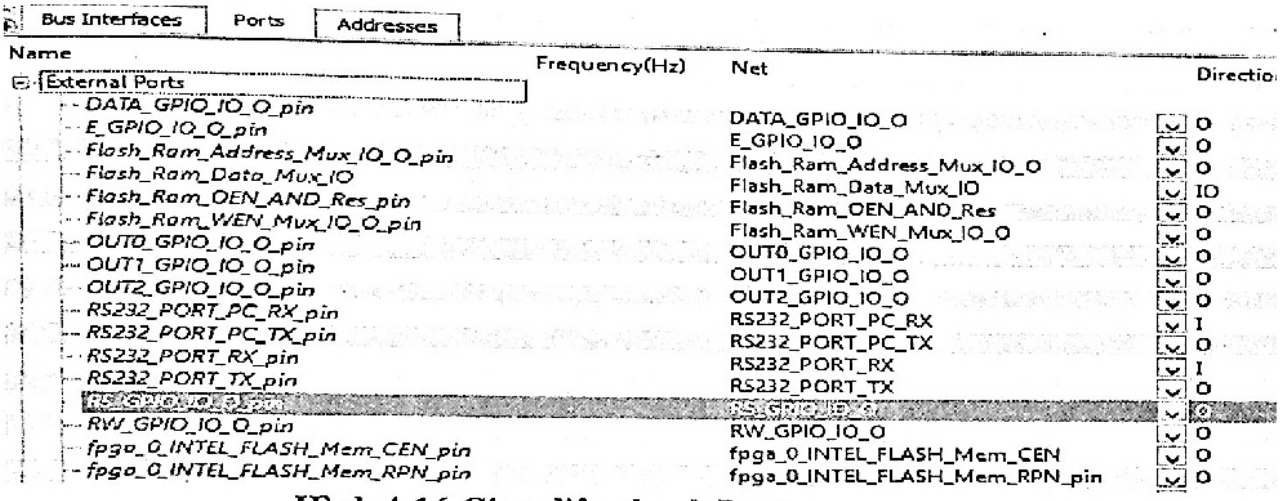
✓ Tương tự như trên thực hiện kết nối đến hệ thống cho các IP còn lại:



Hình 4-15 Giao diện sau khi kết nối tất cả IP đến hệ thống

- **<u>Bước 5</u>: Tạo kết nối Make External**

✓ Chọn thẻ Port.

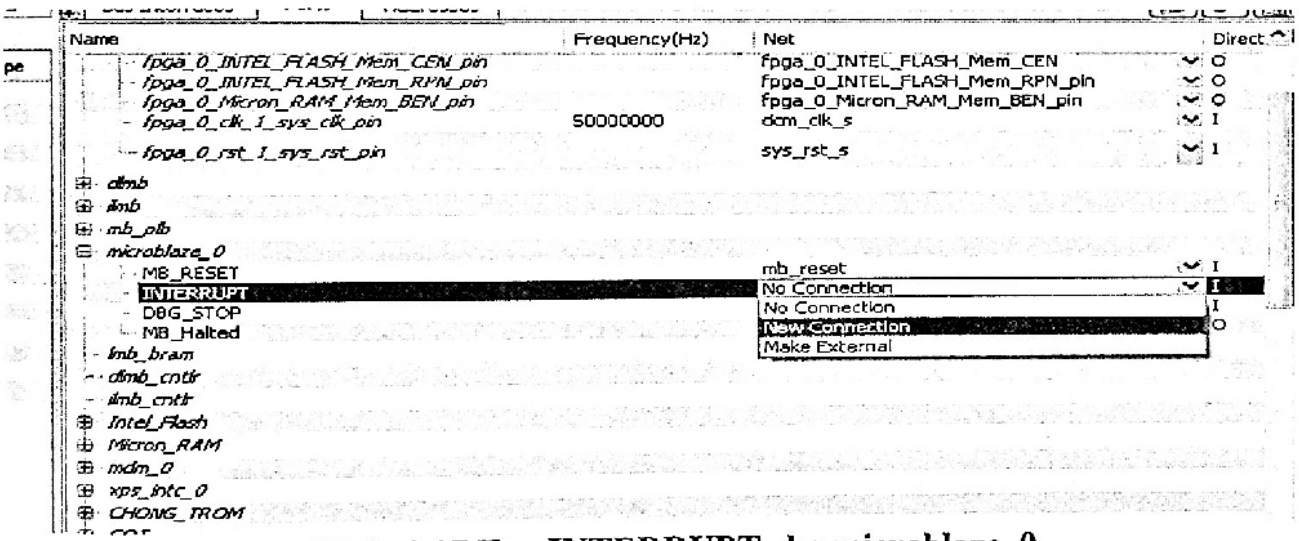


Hình 4-16 Giao diện thanh Port

Kết nối cho xps_intc_0 và delay

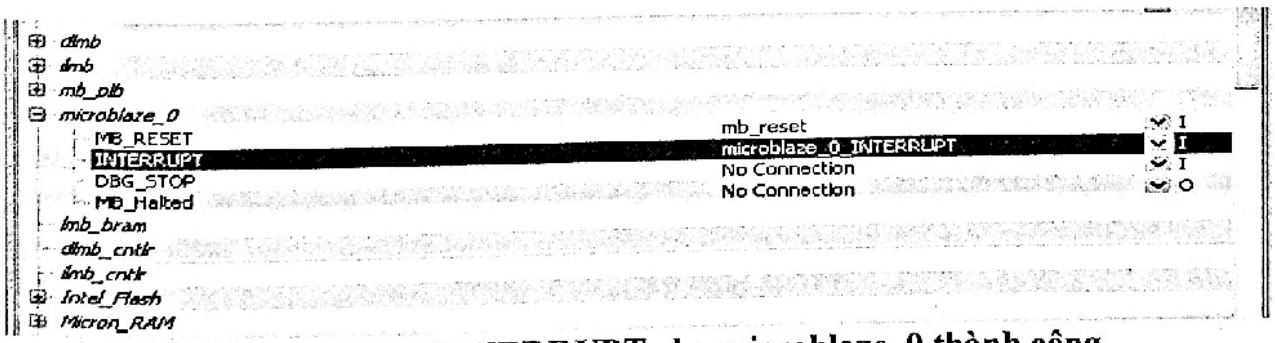
- Mở rộng microblaze_0

✓ **INTERRUPT** chọn **New Connection.**



Hình 4-17 Tạo INTERRUPT cho microblaze_0

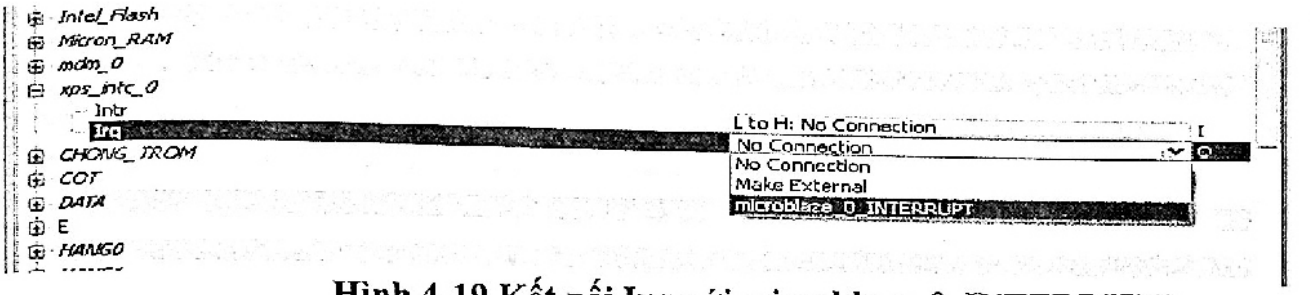
✓ Giao diện sau khi chọn như hình sau:



Hình 4-18 Tạo INTERRUPT cho microblaze_0 thành công

✓ Mở rộng **xps_intc_0.**

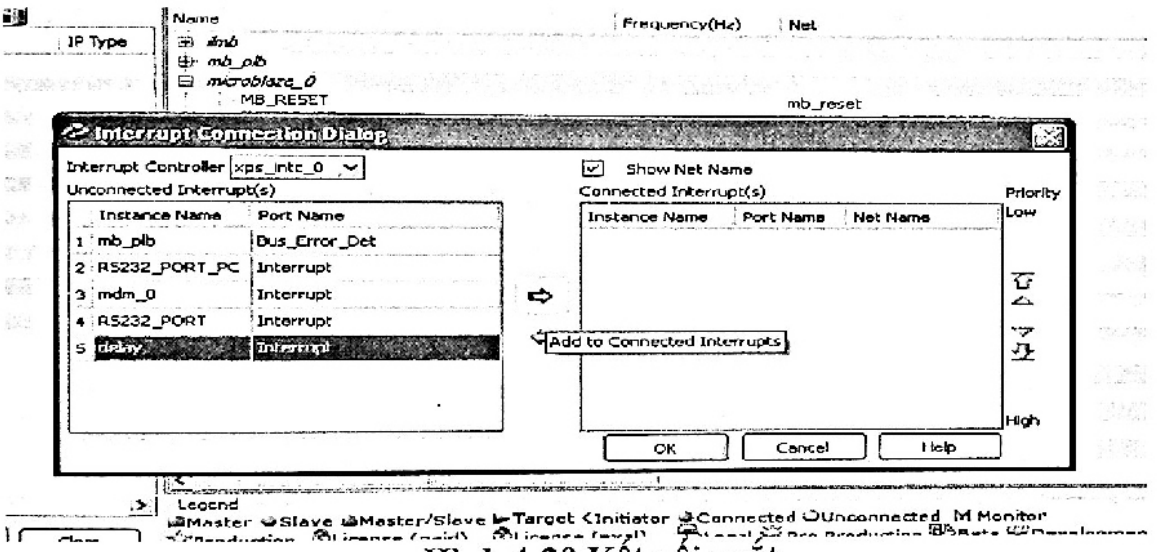
✓ **Irq** chọn **microblaze_0_INTERRUPT.**



Hình 4-19 Kết nối Irq với microblaze_0_INTERRUPT

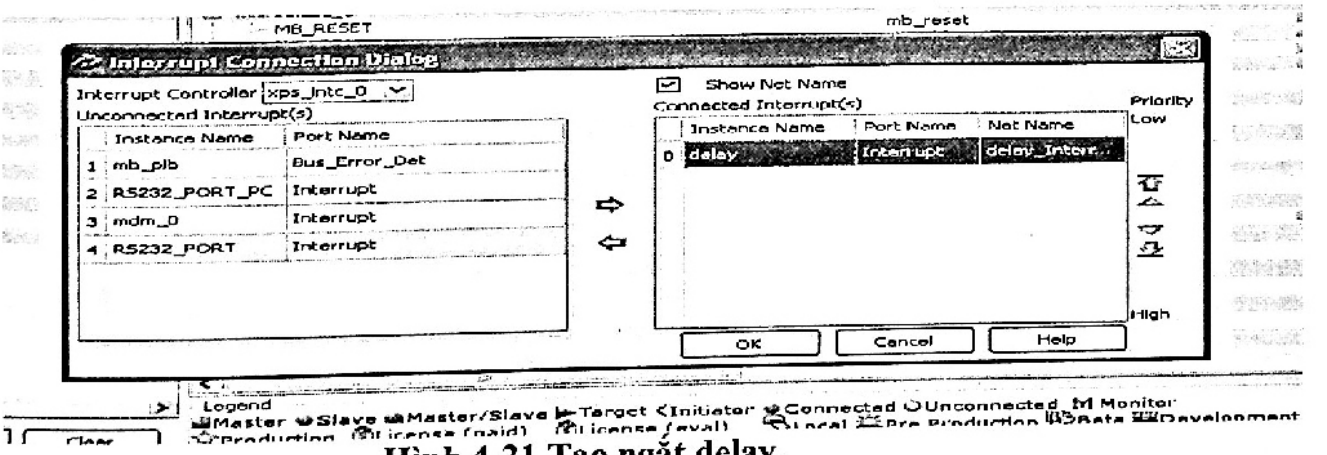
✓ Nhấp chuột vào **l to H: No connection** để mở hộp thoại "Interrupt Connection Dialog".

✓ Chọn **delay** bên khung trái.



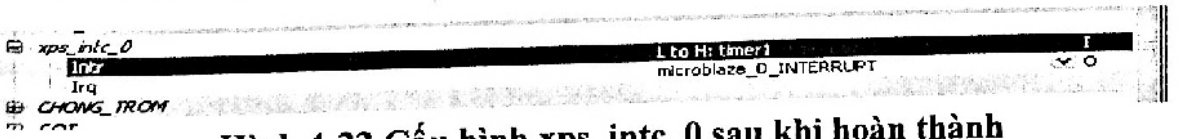
Hình 4-20 Kết nối ngắt

- Nhấn nút ⇨ .



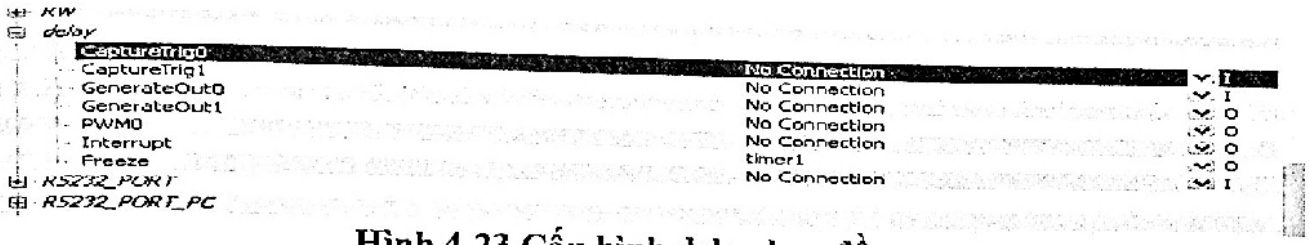
Hình 4-21 Tạo ngắt delay.

- Sửa tên **delay_Interrupt** thành **timer1**
- Ta được xps_intc_0 có cấu hình như sau:



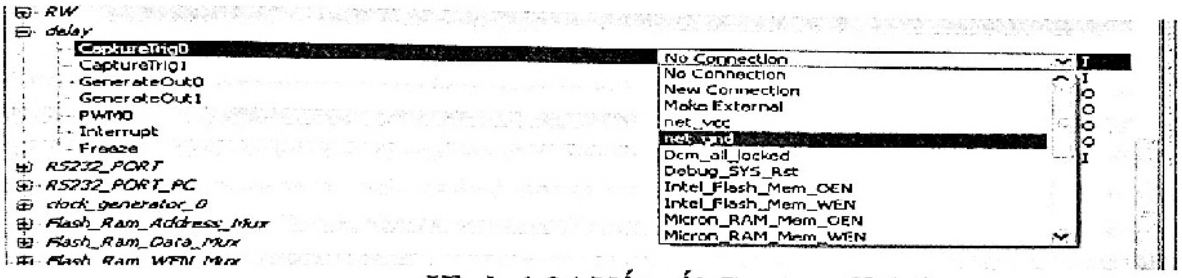
Hình 4-22 Cấu hình xps_intc_0 sau khi hoàn thành

- Mở rộng **delay**



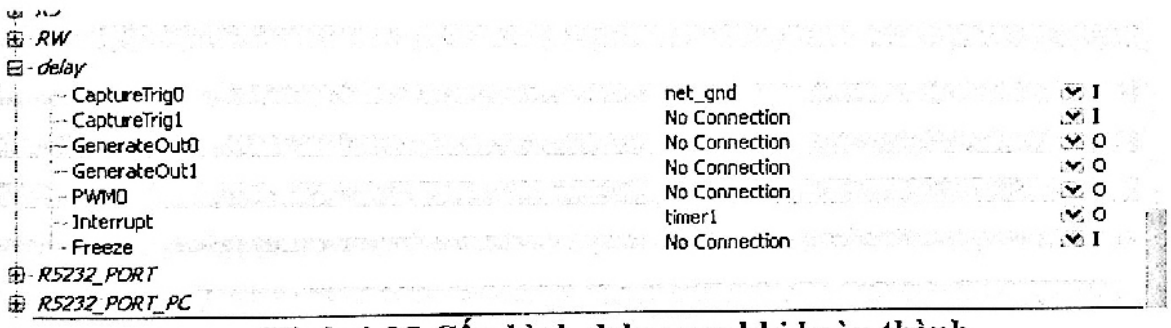
Hình 4-23 Cấu hình delay ban đầu

✓ **CaptureTrig0** chọn **net_gnd.**



Hình 4-24 Kết nối Capture Trig0

✓ Sau khi chọn ta có cấu hình **delay** như sau:



Hình 4-25 Cấu hình delay sau khi hoàn thành

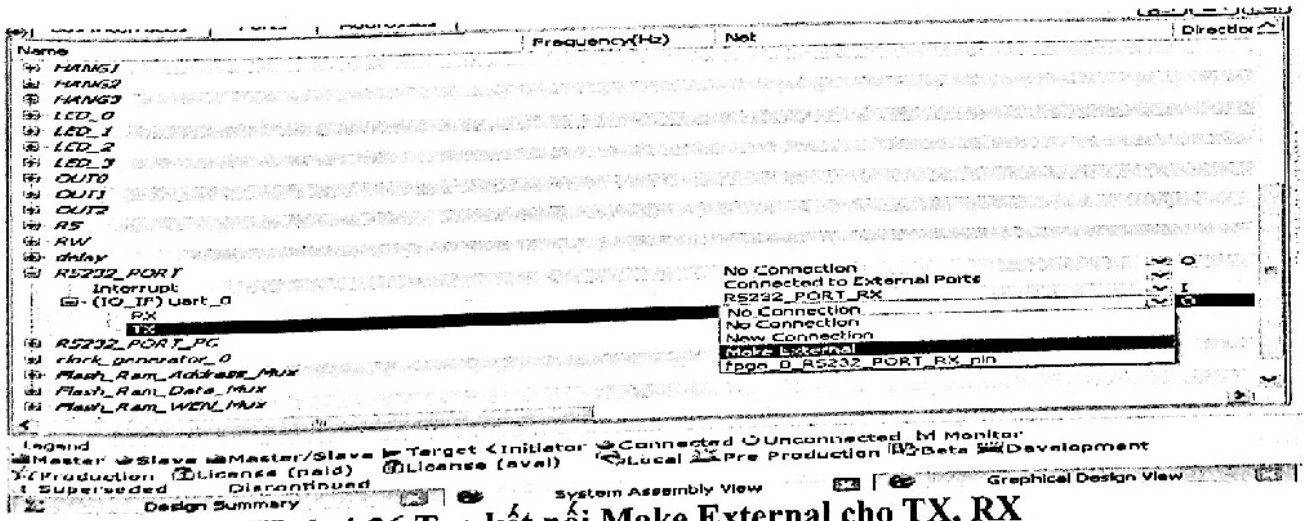
Thực hiện tương tự cho các IP OUT0, OUT1, OUT2, RS, RW, E, DATA.

*Kết nối **Make External** cho **RS232_PORT.***

✓ Mở rộng **RS232_PORT**

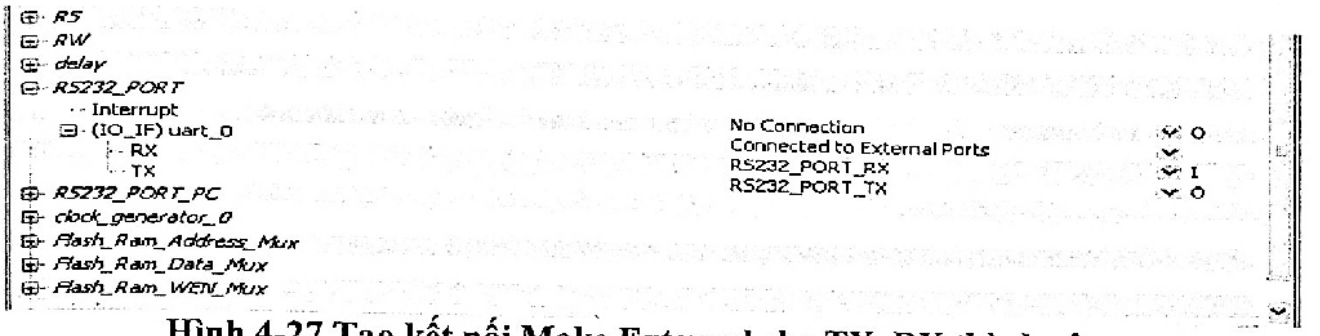
✓ **RX** chọn **Make External**

✓ **TX** chọn **Make External**



Hình 4-26 Tạo kết nối Make External cho TX, RX

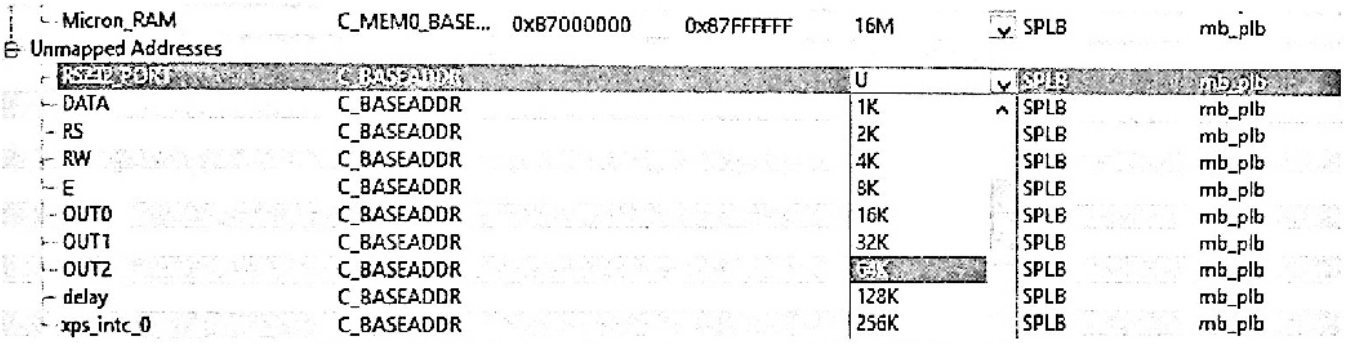
- Ta được như hình sau:



Hình 4-27 Tạo kết nối Make External cho TX, RX thành công

- Thực hiện tương tự với RS232_PORT_PC

- **<u>Bước 6:</u> Chọn dung lượng và địa chỉ bộ nhớ cho các IP.**
  - ✓ Mở rộng **Unmapped Addresses**
  - ✓ Chọn **RS232_PORT** dung lượng bộ nhớ 64K

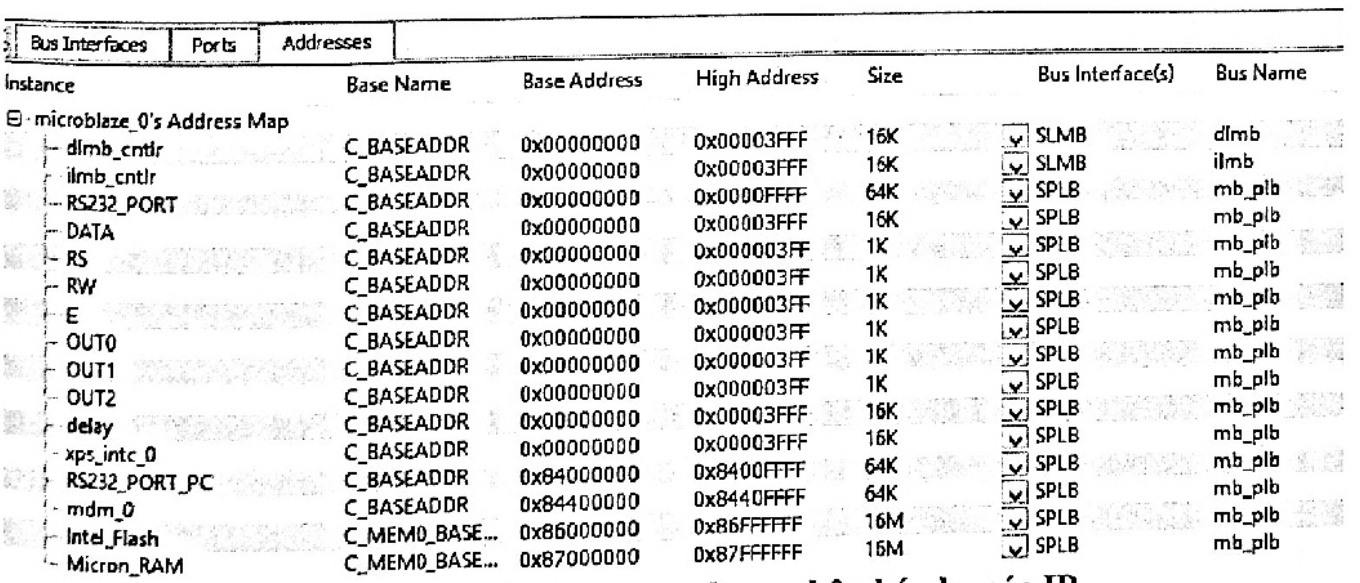


Hình 4-28 Chọn dung lượng và địa chỉ bộ nhớ

✓ Thực hiên tương tự như trên :

- RS232_PORT_PC chọn dung lượng bộ nhớ 64K
- OUT0, OUT1, OUT2, E, RS, RW chọn dung lượng bộ nhớ 1K
- Xps_intc_0, delay, DATA chọn dung lượng bộ nhớ 16K
- Dlmp_cntlr,ilmp_cntlr chọn dung lượng bộ nhớ 32K

✓ Kết thúc các giai đoạn trên ta có hình sau:



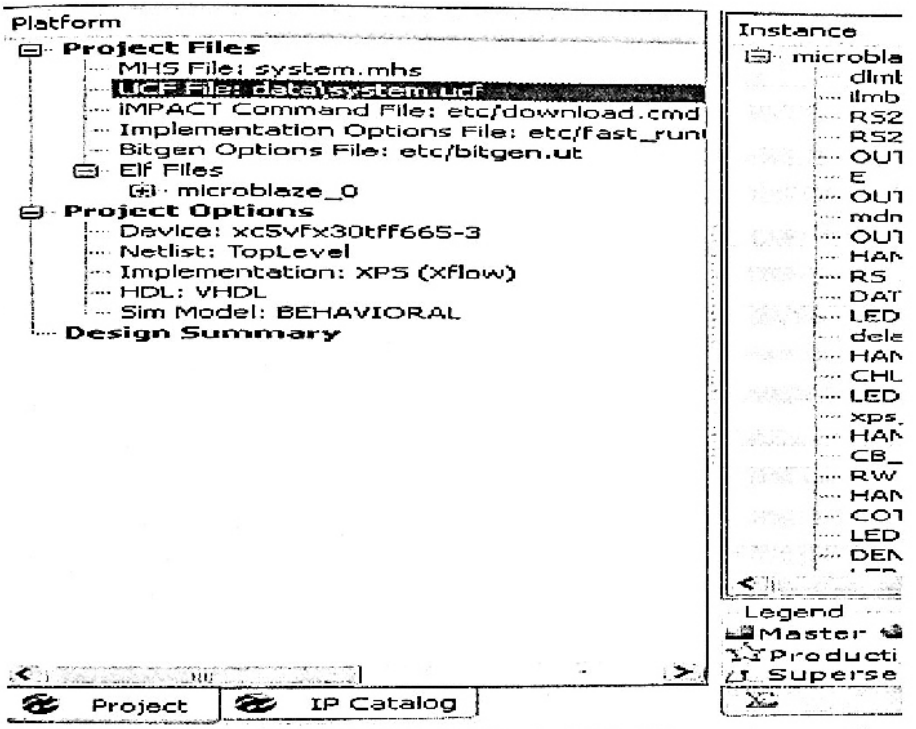
Hình 4-29 Sau khi chọn dung lượng bộ nhớ cho các IP

✓ Chọn để sắp xếp địa chỉ hợp lí:

- **<u>Bước 7</u> : Gán chân cho các IP**

✓ Chọn thẻ **Project**

✓ Nhấp mở file **system.ucf**



Hình 4-30 File system.ucf

Soạn đoạn văn bản dưới sau ##### **Nexys 2-500** sau để gán chân các IP và lưu lại

```
# Nexys 2-500 Board
Net OUT0_GPIO_IO_O_pin<0> LOC=M15 | IOSTANDARD = LVCMOS33;
Net OUT1_GPIO_IO_O_pin<0> LOC=K12 | IOSTANDARD = LVCMOS33;
Net OUT2_GPIO_IO_O_pin<0> LOC=L17 | IOSTANDARD = LVCMOS33;

Net DATA_GPIO_IO_O_pin<7> LOC=G15 | IOSTANDARD = LVCMOS33;
Net DATA_GPIO_IO_O_pin<6> LOC=J16 | IOSTANDARD = LVCMOS33;
Net DATA_GPIO_IO_O_pin<5> LOC=G13 | IOSTANDARD = LVCMOS33;
Net DATA_GPIO_IO_O_pin<4> LOC=H16 | IOSTANDARD = LVCMOS33;
Net DATA_GPIO_IO_O_pin<3> LOC=H15 | IOSTANDARD = LVCMOS33;
Net DATA_GPIO_IO_O_pin<2> LOC=F14 | IOSTANDARD = LVCMOS33;
Net DATA_GPIO_IO_O_pin<1> LOC=G16 | IOSTANDARD = LVCMOS33;
Net DATA_GPIO_IO_O_pin<0> LOC=J12 | IOSTANDARD = LVCMOS33;

Net E_GPIO_IO_O_pin<0> LOC=J15 | IOSTANDARD = LVCMOS33;
Net RS_GPIO_IO_O_pin<0> LOC=K14 | IOSTANDARD = LVCMOS33;
Net RW_GPIO_IO_O_pin<0> LOC=K15 | IOSTANDARD = LVCMOS33;

Net RS232_PORT_RX_pin LOC=M14 | IOSTANDARD = LVCMOS33 | PULLUP;
Net RS232_PORT_TX_pin LOC=M16 | IOSTANDARD = LVCMOS33 | DRIVE = 2;
```

```
Net RS232_PORT_PC_RX_pin LOC=U6 | IOSTANDARD = LVCMOS33 |
PULLUP;
Net RS232_PORT_PC_TX_pin LOC=P9 | IOSTANDARD = LVCMOS33 | DRIVE =
2;

Net Flash_Ram_Address_Mux_IO_O_pin<22> LOC=J1 | IOSTANDARD =
LVCMOS33 | DRIVE=2;
Net Flash_Ram_Address_Mux_IO_O_pin<21> LOC=J2 | IOSTANDARD =
LVCMOS33 | DRIVE=2;
Net Flash_Ram_Address_Mux_IO_O_pin<20> LOC=H4 | IOSTANDARD =
LVCMOS33 | DRIVE=2;
Net Flash_Ram_Address_Mux_IO_O_pin<19> LOC=H1 | IOSTANDARD =
LVCMOS33 | DRIVE=2;
Net Flash_Ram_Address_Mux_IO_O_pin<18> LOC=H2 | IOSTANDARD =
LVCMOS33 | DRIVE=2;
Net Flash_Ram_Address_Mux_IO_O_pin<17> LOC=J5 | IOSTANDARD =
LVCMOS33 | DRIVE=2;
Net Flash_Ram_Address_Mux_IO_O_pin<16> LOC=H3 | IOSTANDARD =
LVCMOS33 | DRIVE=2;
Net Flash_Ram_Address_Mux_IO_O_pin<15> LOC=H6 | IOSTANDARD =
LVCMOS33 | DRIVE=2;
Net Flash_Ram_Address_Mux_IO_O_pin<14> LOC=F1 | IOSTANDARD =
LVCMOS33 | DRIVE=2;
Net Flash_Ram_Address_Mux_IO_O_pin<13> LOC=G3 | IOSTANDARD =
LVCMOS33 | DRIVE=2;
Net Flash_Ram_Address_Mux_IO_O_pin<12> LOC=G6 | IOSTANDARD =
LVCMOS33 | DRIVE=2;
Net Flash_Ram_Address_Mux_IO_O_pin<11> LOC=G5 | IOSTANDARD =
LVCMOS33 | DRIVE=2;
Net Flash_Ram_Address_Mux_IO_O_pin<10> LOC=G4 | IOSTANDARD =
LVCMOS33 | DRIVE=2;
Net Flash_Ram_Address_Mux_IO_O_pin<9> LOC=F2 | IOSTANDARD =
LVCMOS33 | DRIVE=2;
Net Flash_Ram_Address_Mux_IO_O_pin<8> LOC=E1 | IOSTANDARD =
LVCMOS33 | DRIVE=2;
Net Flash_Ram_Address_Mux_IO_O_pin<7> LOC=M5 | IOSTANDARD =
LVCMOS33 | DRIVE=2;
Net Flash_Ram_Address_Mux_IO_O_pin<6> LOC=E2 | IOSTANDARD =
LVCMOS33 | DRIVE=2;
Net Flash_Ram_Address_Mux_IO_O_pin<5> LOC=C2 | IOSTANDARD =
LVCMOS33 | DRIVE=2;
Net Flash_Ram_Address_Mux_IO_O_pin<4> LOC=C1 | IOSTANDARD =
LVCMOS33 | DRIVE=2;
Net Flash_Ram_Address_Mux_IO_O_pin<3> LOC=D2 | IOSTANDARD =
LVCMOS33 | DRIVE=2;
Net Flash_Ram_Address_Mux_IO_O_pin<2> LOC=K3 | IOSTANDARD =
```

```
LVCMOS33 | DRIVE=2;
Net Flash_Ram_Address_Mux_IO_O_pin<1> LOC=D1 | IOSTANDARD =
LVCMOS33 | DRIVE=2;
Net Flash_Ram_Address_Mux_IO_O_pin<0> LOC=K6 | IOSTANDARD =
LVCMOS33 | DRIVE=2;
Net Flash_Ram_OEN_AND_Res_pin<0> LOC=T2 | IOSTANDARD = LVCMOS33 |
DRIVE=2 | PULLUP;
Net Flash_Ram_WEN_Mux_IO_O_pin<0> LOC=N7 | IOSTANDARD = LVCMOS33
| DRIVE=2;

Net fpga_0_Micron_RAM_Mem_BEN_pin<0> LOC=K4 | IOSTANDARD =
LVCMOS33 | DRIVE=2 | PULLDOWN;
Net fpga_0_Micron_RAM_Mem_BEN_pin<1> LOC=K5 | IOSTANDARD =
LVCMOS33 | DRIVE=2 | PULLDOWN;

Net fpga_0_INTEL_FLASH_Mem_RPN_pin LOC=T5 | IOSTANDARD =
LVCMOS33 | DRIVE=2 | PULLUP;
Net fpga_0_INTEL_FLASH_Mem_CEN_pin LOC=R5 | IOSTANDARD =
LVCMOS33 | DRIVE=2 | PULLUP;

Net Flash_Ram_Data_Mux_IO<15> LOC=L1 | IOSTANDARD = LVCMOS33 |
DRIVE=2;
Net Flash_Ram_Data_Mux_IO<14> LOC=L4 | IOSTANDARD = LVCMOS33 |
DRIVE=2;
Net Flash_Ram_Data_Mux_IO<13> LOC=L6 | IOSTANDARD = LVCMOS33 |
DRIVE=2;
Net Flash_Ram_Data_Mux_IO<12> LOC=M4 | IOSTANDARD = LVCMOS33 |
DRIVE=2;
Net Flash_Ram_Data_Mux_IO<11> LOC=N5 | IOSTANDARD = LVCMOS33 |
DRIVE=2;
Net Flash_Ram_Data_Mux_IO<10> LOC=P1 | IOSTANDARD = LVCMOS33 |
DRIVE=2;
Net Flash_Ram_Data_Mux_IO<9> LOC=P2 | IOSTANDARD = LVCMOS33 |
DRIVE=2;
Net Flash_Ram_Data_Mux_IO<8> LOC=R2 | IOSTANDARD = LVCMOS33 |
DRIVE=2;
Net Flash_Ram_Data_Mux_IO<7> LOC=L3 | IOSTANDARD = LVCMOS33 |
DRIVE=2;
Net Flash_Ram_Data_Mux_IO<6> LOC=L5 | IOSTANDARD = LVCMOS33 |
DRIVE=2;
Net Flash_Ram_Data_Mux_IO<5> LOC=M3 | IOSTANDARD = LVCMOS33 |
DRIVE=2;
Net Flash_Ram_Data_Mux_IO<4> LOC=M6 | IOSTANDARD = LVCMOS33 |
DRIVE=2;
Net Flash_Ram_Data_Mux_IO<3> LOC=L2 | IOSTANDARD = LVCMOS33 |
DRIVE=2;
Net Flash_Ram_Data_Mux_IO<2> LOC=N4 | IOSTANDARD = LVCMOS33 |
```

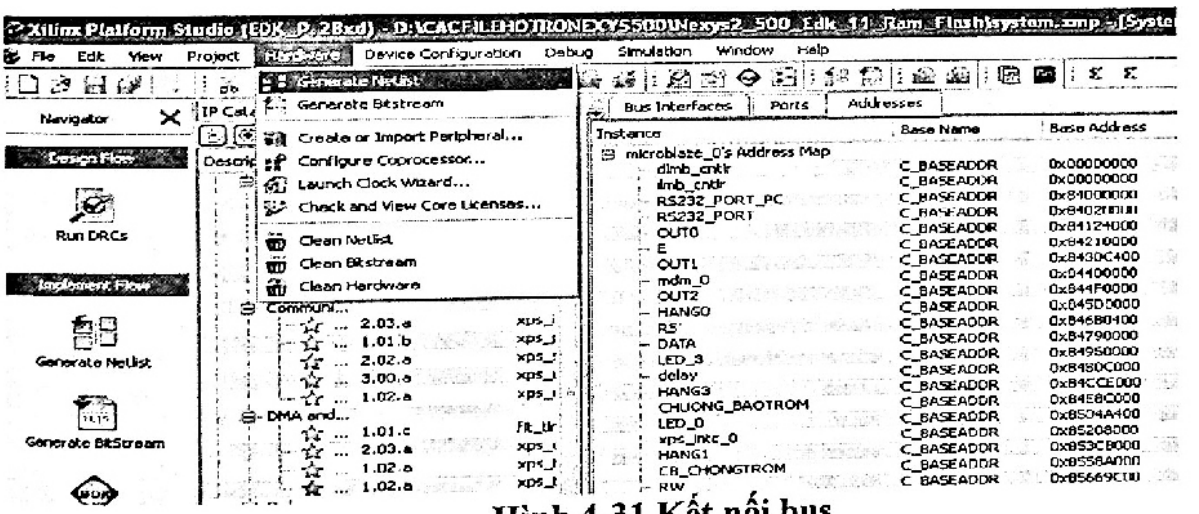
```
DRIVE=2;
Net Flash_Ram_Data_Mux_IO<1> LOC=R3 | IOSTANDARD = LVCMOS33 |
DRIVE=2;
Net Flash_Ram_Data_Mux_IO<0> LOC=T1 | IOSTANDARD = LVCMOS33 |
DRIVE=2;

Net fpga_0_clk_1_sys_clk_pin TNM_NET = sys_clk_pin;
TIMESPEC TS_sys_clk_pin = PERIOD sys_clk_pin 50000 kHz;
Net fpga_0_clk_1_sys_clk_pin LOC=B8 | IOSTANDARD = LVCMOS33;
Net fpga_0_rst_1_sys_rst_pin TIG;
Net fpga_0_rst_1_sys_rst_pin LOC=B18 | IOSTANDARD = LVCMOS33;
```

✓ Nhấn để lưu lại các cài đặt.

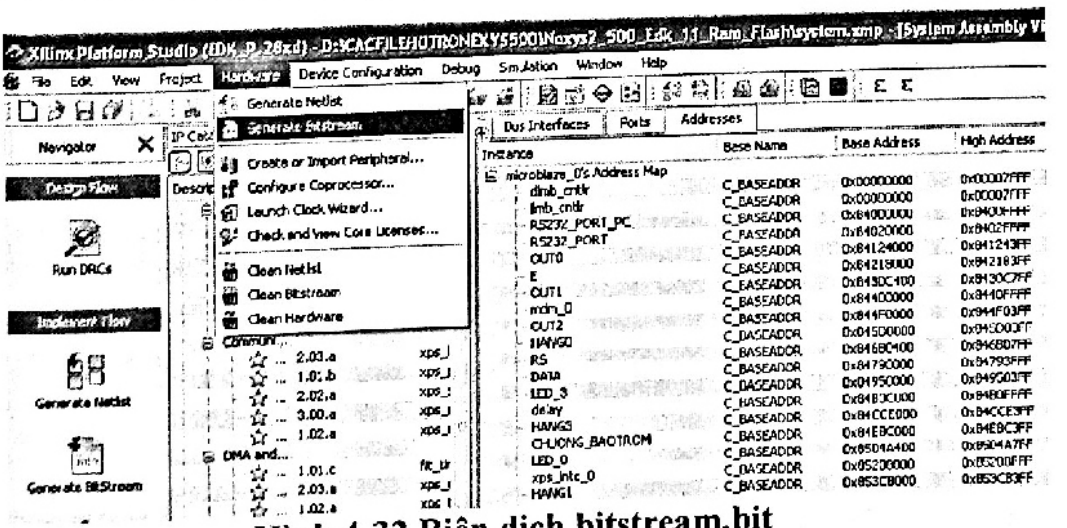
- **<u>Bước 8</u> : Tạo kết nối và bitstream**

✓ Chọn **Hardware -> Generate Netlist** để kết nối dây phần cứng và đợi cho đến khi kết nối xong



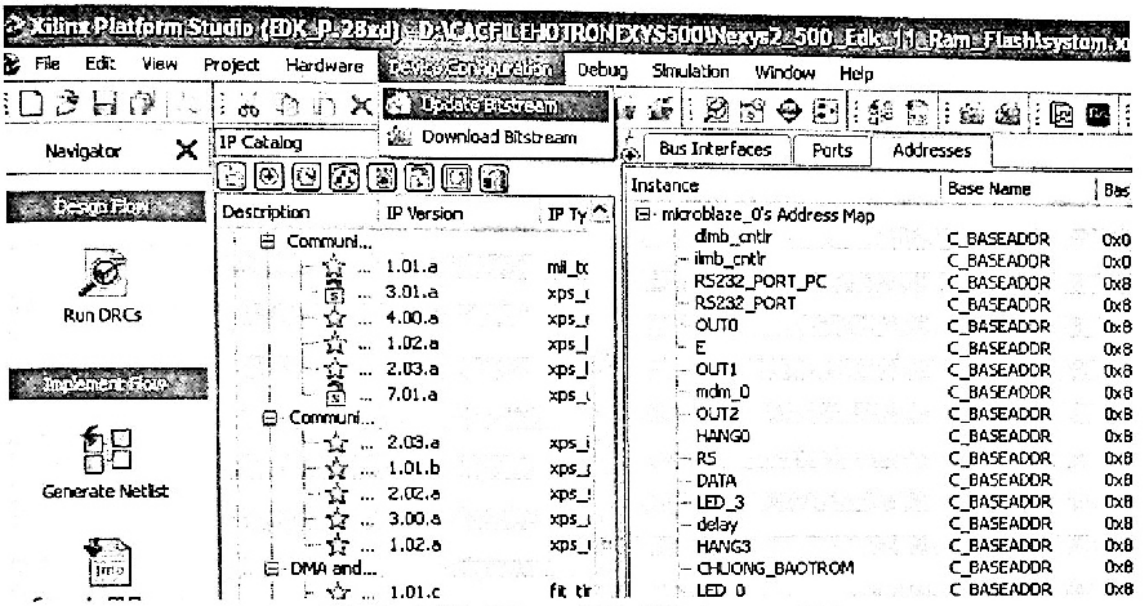
Hình 4-31 Kết nối bus

✓ Chọn **Hardware -> Generate Bitstream** để biên dịch ra file bitstream.bit.



Hình 4-32 Biên dịch bitstream.bit

✓ Chọn **Device Configuration -> Update Bitstream**



Hình 4-33 Cập nhật Bitstream.bit

4.3 THIẾT KẾ HỆ THỐNG NHÚNG TRÊN FPGA DÙNG SDK:

4.3.1 Xây dựng các yêu cầu điều khiển theo tin nhắn

Để điều khiển thiết bị tắt mở, người điều khiển soạn tin theo cú pháp MK12345678 On1

- Trong đó 12345678 là mật khẩu yêu cầu của hệ thống
- On1 là chế độ mở của thiết bị cần điều khiển. Ngoài ra có các chế độ khác như On2, On3, Off1, Off2, Off3 ứng với 3 thiết bị.
- Người điều khiển cần nhập đúng mật khẩu đúng chế độ yêu cầu của thiết bị tương ứng và cú pháp điều khiển để điều khiển các thiết bị điện.
- Khi có tin nhắn gửi tới module sim, module sim sẽ truyền dữ liệu đến FPGA báo có tin nhắn gửi tới. FPGA sẽ đưa các lệnh yêu cầu đọc tin nhắn từ module sim và sẽ xử lí tin nhắn .Nếu tin nhắn đúng với cú pháp như trên thì hệ thống sẽ gửi tin nhắn thông báo cho người điều khiển thiết bị nào đang tắt thiết bị nào đang mở.
- LCD có nhiệm vụ hiển thị thiết bị nào tắt mở

4.3.2 Lưu đồ chương trình

Để viết lưu đồ thì cần phải biết các bước thực hiện của hệ thống điều khiển.

Hệ thống sau khi cấp điện sẽ thực hiện các công việc sau:

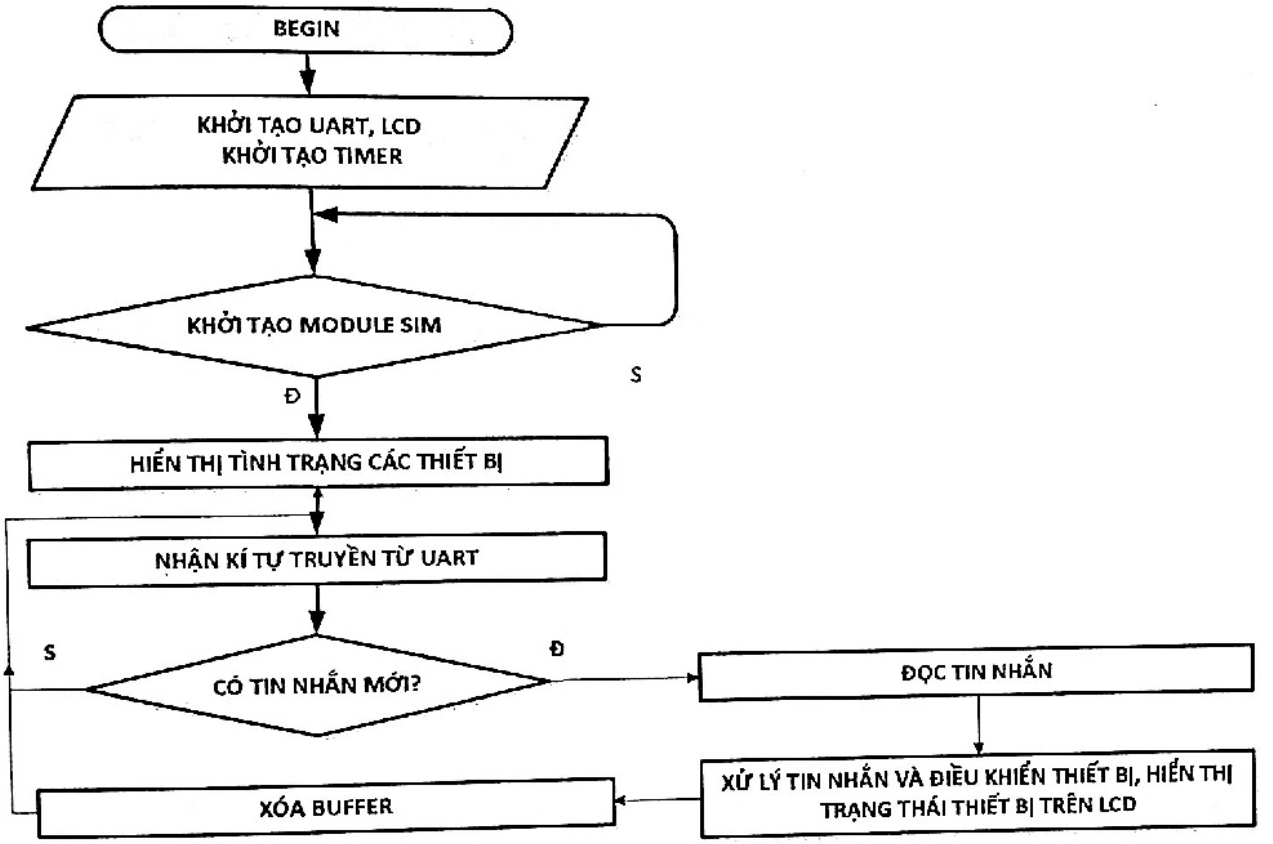
Bước 1: Khởi tạo truyền dữ liệu UART để giao tiếp với khối SMS.

Bước 2: Khởi tạo LCD để hiển thị các thông tin.

Bước 3: Đọc mã điều khiển thiết bị đã lưu trong bộ nhớ để điều khiển thiết bị.

Khi có tin nhắn do bộ nhận dữ liệu tin nhắn từ khối SMS tạo ra thì tiến hành nhận dữ liệu và xử lý theo yêu cầu của tin nhắn rồi trở lại chương trình chính thực hiện tiếp.

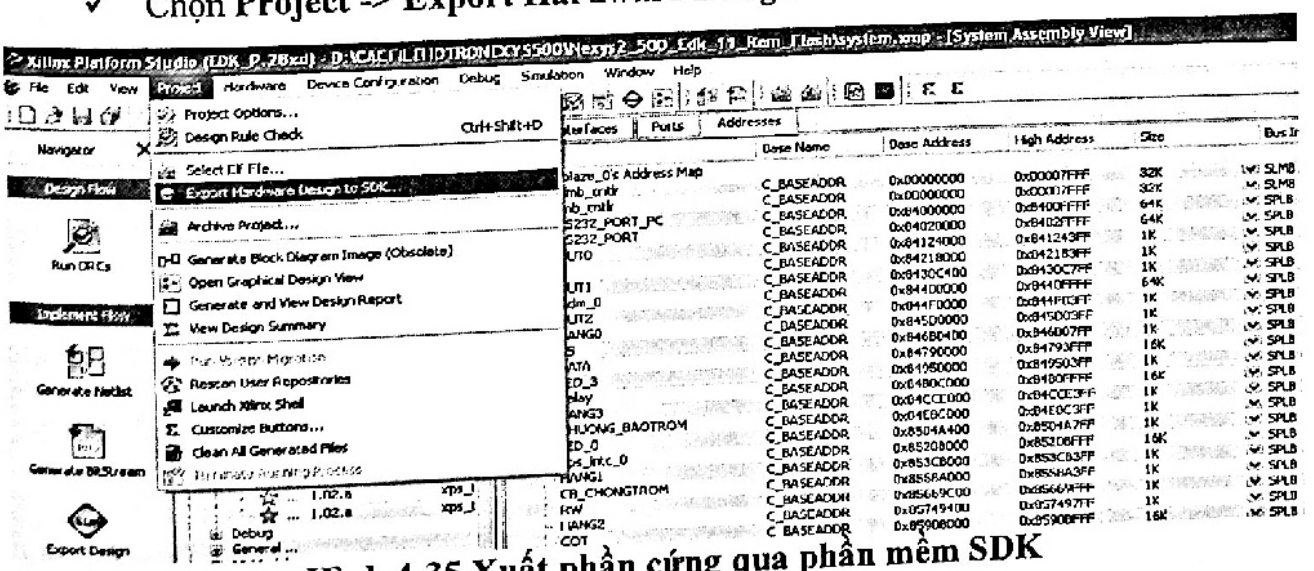
➢ **Lưu đồ chương trình chính**



Hình 4-34 Lưu đồ chương trình chính

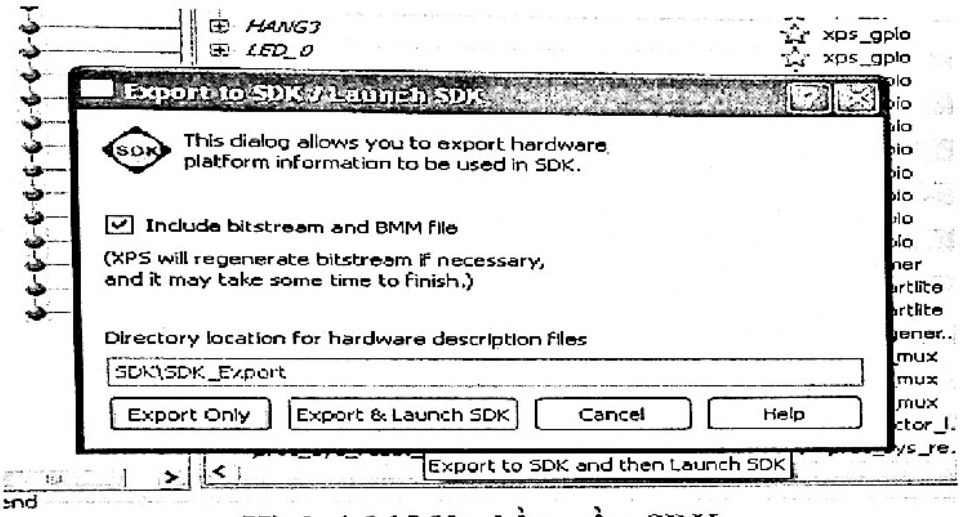
4.3.3 Xây dựng phần mềm:

✓ Chọn **Project -> Export Hardware Design to SDK**



Hình 4-35 Xuất phần cứng qua phần mềm SDK

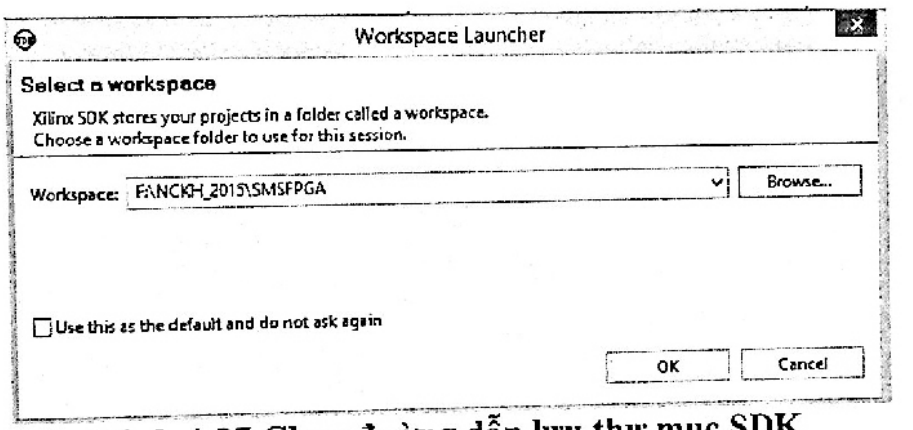
✓ Chọn **Export & Launch SDK.**



Hình 4-36 Mở phần mềm SDK

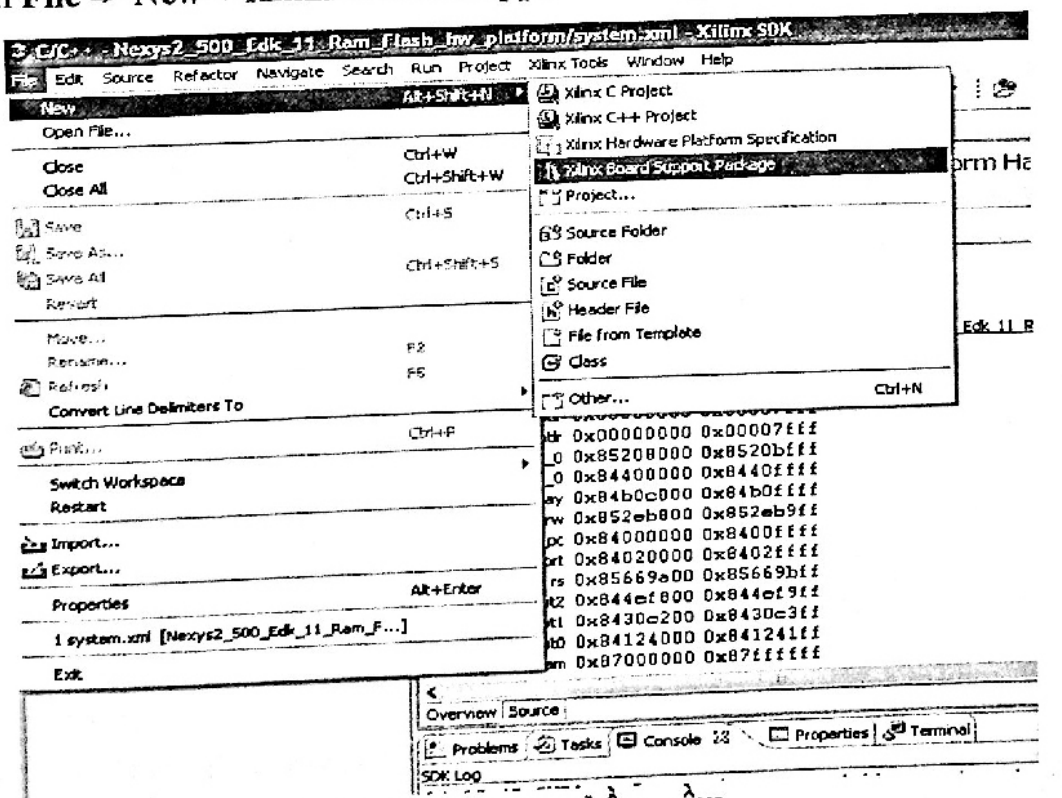
✓ Chờ cho đến khi phần mềm SDK được mở lên.

✓ Chọn đường dẫn lưu thư mục SDK.



Hình 4-37 Chọn đường dẫn lưu thư mục SDK

✓ Chọn **File -> New ->Xilinx Board Support Package.**



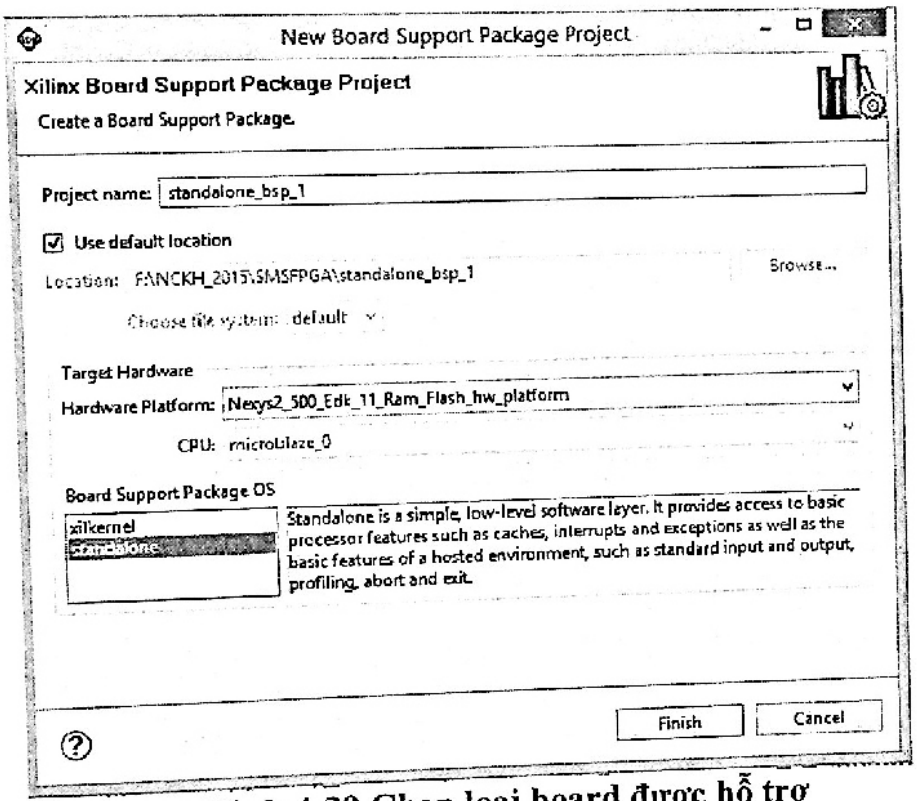
Hình 4-38 Tạo phần mềm

✓ Hộp thoại "Xilinx Board Support Package Project"

Có 2 lựa chọn :

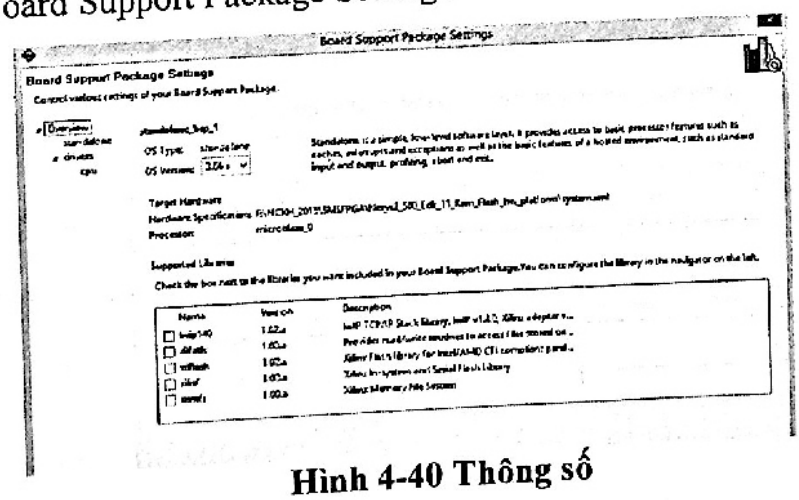
- **xilkernel**: hỗ trợ ở mức độ cao cung cấp POSIX với bộ lập lịch, đồng bộ, timer như là một ngoại vi kết nối với vi xử lý.
- **standalone**: hỗ trợ ở mức độ đơn giản bào gồm các kết nối như : bộ nhớ đệm, ngắt, các port xuất nhập I/O.

✓ Chọn **standalone**



Hình 4-39 Chọn loại board được hỗ trợ

✓ Hộp thoại "Board Support Package Settings"

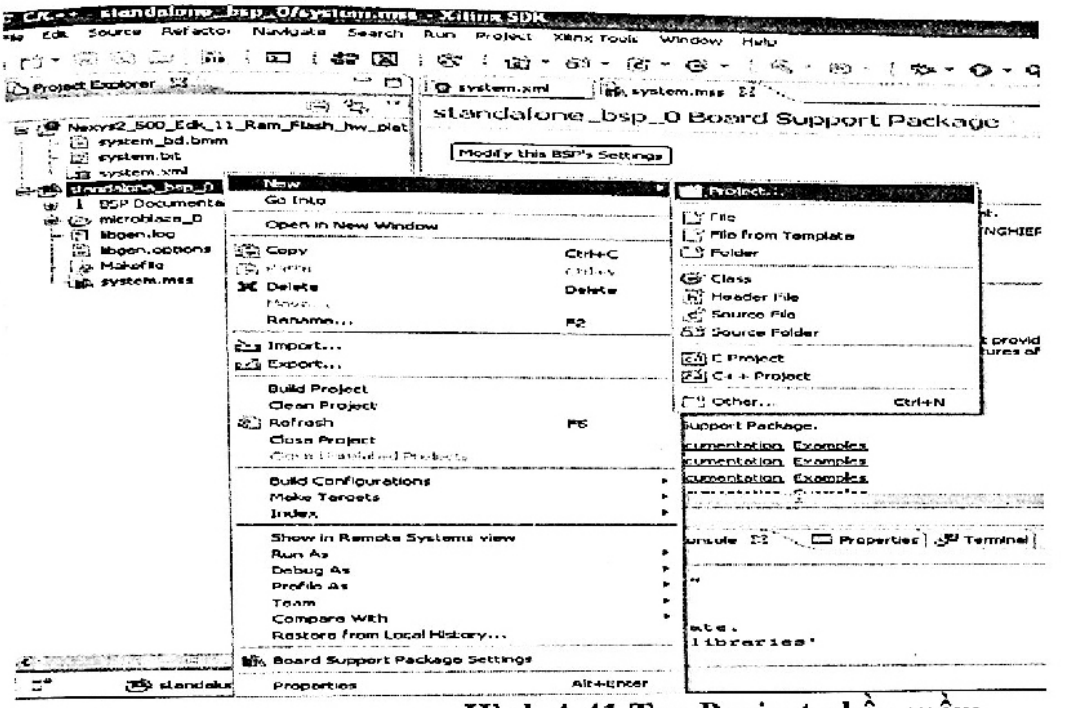


Hình 4-40 Thông số

✓ Cửa sổ **Project Explorer**

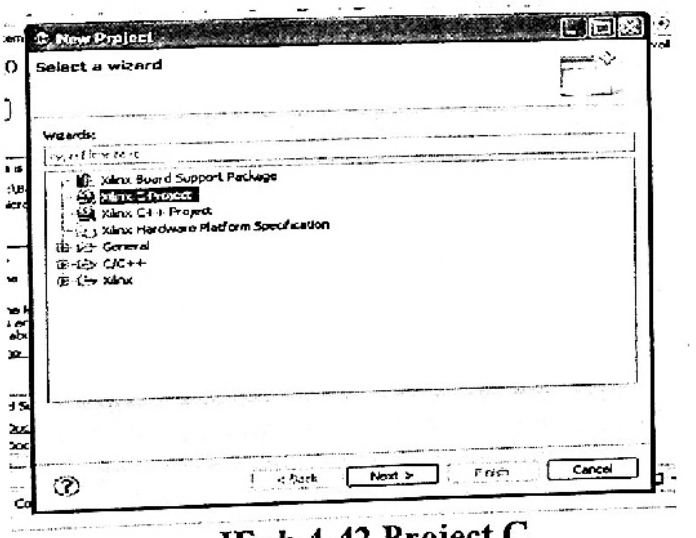
✓ Nhấp phải chuột **standalone_bsp_0**

✓ Chọn **New-> Project.**



Hình 4-41 Tạo Project phần mềm

- ✓ Cửa sổ **New Project.**
- ✓ Chọn **Xilinx C Project.**
- ✓ Chọn **Next.**



Hình 4-42 Project C

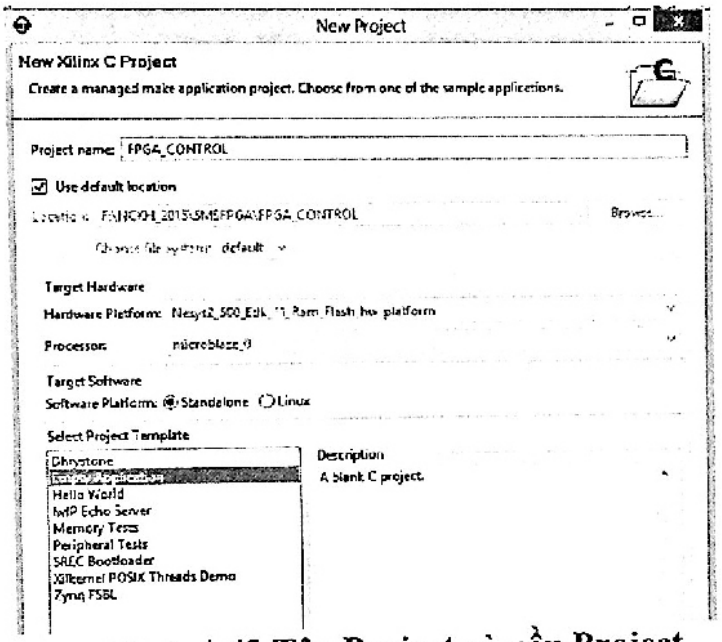
- ✓ Cửa sổ **New Xilinx C Project**
- ✓ **Project Name(1)** : đặt tên Project là FPGA_CONTROL
- ✓ **Project Template(2)** : nền cho phần mềm

Có 8 lựa chọn :

- **Dhrystone:** chương trình đánh giá phần cứng
- **Empty Application:** 1 file C rỗng
- **Hello World:** chương trình hello world
- **iwIP Echo Server:** chương trình iwIP Echo ứng dụng web

- **Memory Tests:** chương trình kiểm tra bộ nhớ
- **Peripheral Tests:** kiểm tra ngoại vi
- **SREC Bootloader:** chương trình SREC bootloader
- **Xilkernel POSIX Threads Demo:** chương trình ví dụ POSIX

✓ Chọn **Empty Application.**



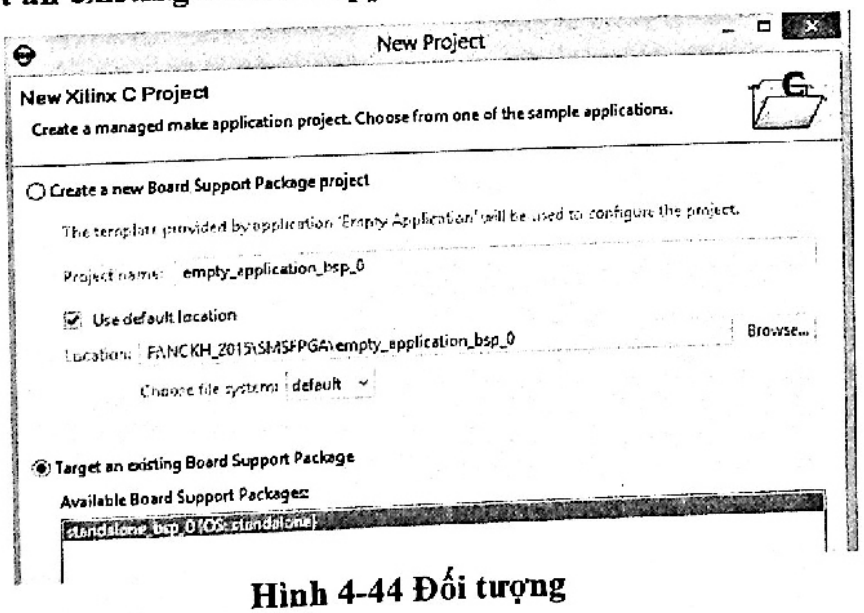
Hình 4-43 Tên Project và nền Project

✓ Chọn Next.

Có 2 lựa chọn :

- **Create a new Board Support Package Project(1):** tạo mới cho 1 nền phần cứng mới
- **Target an existing Board Support Package(2):** tạo cho gói nền phần cứng có sẵn

✓ Chọn **Target an existing Board Support Package .**

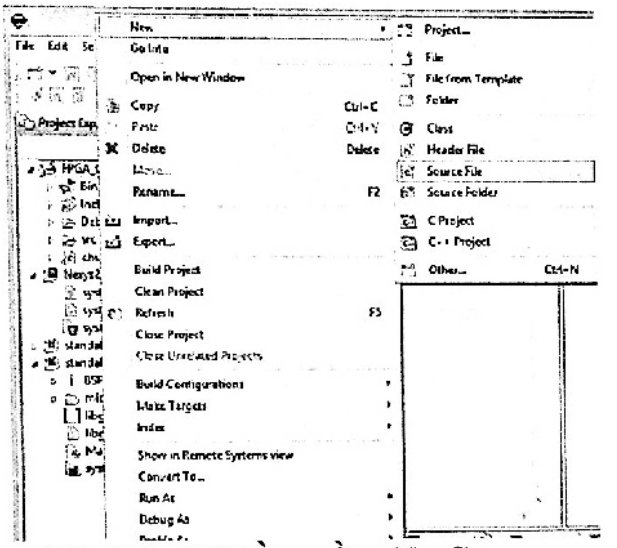


Hình 4-44 Đối tượng

✓ Chọn Finish.

✓ Nhấp chuột phải folder

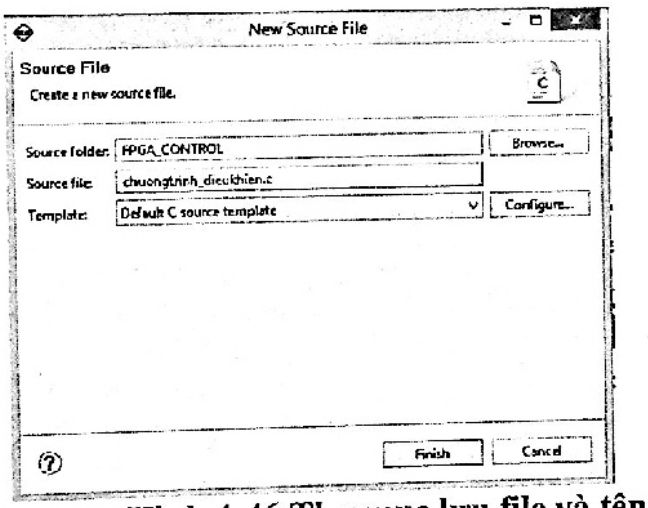
✓ Chọn **File New–>source**



Hình 4-45 Phần mềm file C

✓ **Source folder**: chọn folder FPGA_CONTROL

✓ **Soure file :** đặt tên chuongtrinh_dieukhien.c

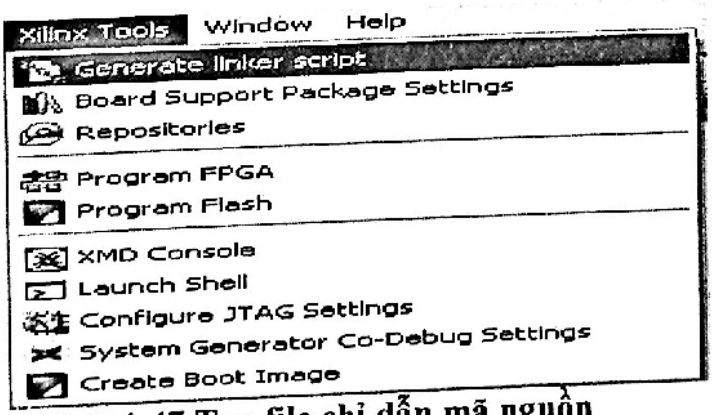


Hình 4-46 Thư mục lưu file và tên file

✓ Chọn Finish.

Mở file **chuongtrinh_dieukhien.c** và thêm vào code (code được chép ra đĩa CD)

✓ Chọn **Xilinx Tools -> Generate linker script.**



Hình 4-47 Tạo file chỉ dẫn mã nguồn

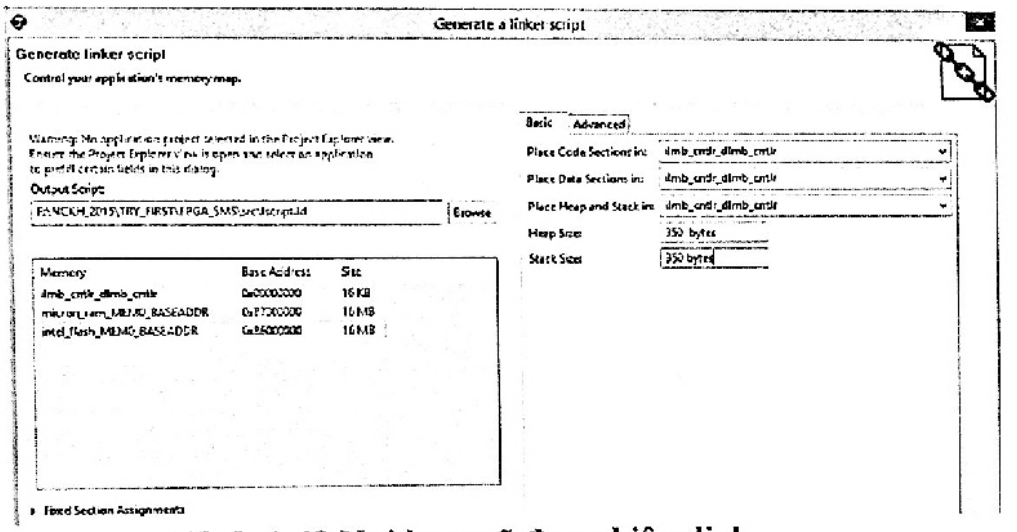
✓ Hộp thoại "Generate linker script"

✓ **Output script**: chọn

F:\NCKH_2015\SMSFPGA\FPGA_CONTROL\src\lscript.ld

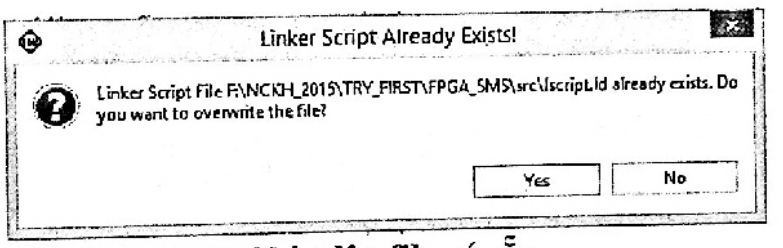
✓ **Heap Size**: 350 bytes

✓ **Stack Size**: 350 bytes



Hình 4-48 Nơi lưu mã được biên dịch

✓ Chọn Generate.

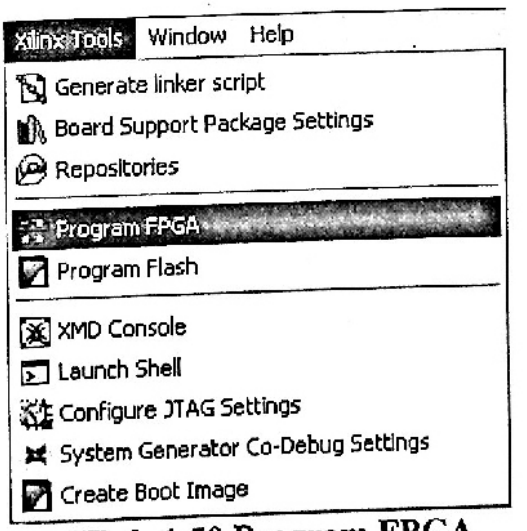


Hình 4-49 Chèn lên file có sẵn

✓ Chọn Yes để ghi chèn lên file lscript.ld đã có sẵn.

✓ Biên dịch và nạp cho FPGA.

Chọn **Xilinx Tools-> Program FPGA**



Hình 4-50 Program FPGA

KẾT LUẬN

Nhóm nghiên cứu đã thực hiện đúng các các yêu cầu đề ra, mạch chạy ổn định.

Trong quá trình thực hiện nhóm tác giả đã xây dựng được cấu hình cho chip FPGA, Spartan 3E, gồm một lõi vi điều khiển và các IP là: một cổng uart để giao tiếp với module sim 900, một timer, một interrupt, các cổng GPIO để giao tiếp với LCD và các thiết bị công suất . Đồng thời cũng xây dựng chương trình phần mềm để điều khiển.

Nhóm đã hoàn thành đề tài và rất mong nhận sự đóng góp thêm từ các phản biện để đề tài hoàn thiện thêm.

KIẾN NGHỊ VÀ HƯỚNG PHÁT TRIỂN

Đề tài mở ra việc xem xét sử dụng FPGA để thay thế vi điều khiển trong một số ứng dụng như: xử lý ảnh, nhận dạng tiếng nói, xử lý số tín hiệu.....và nghiên cứu FPGA với thuật toán xử lý tín hiệu trong matlab và Simulink.

TÀI LIỆU THAM KHẢO

1. Dr Peter R. Wilson , Design Recipes for FPGAs
2. Digilent Nexsy2 board reference manual, june 21,2008.
3. Newnes.Digital.Systems.Design.with.FPGAs.and.CPLDs
4. Nguyễn Đình Phú, Kỹ thuật pld và asic, ĐHSPKT HCM, 2012
5. Nguyễn Đình Phú, Tài liệu thực hành pld và asic, ĐHSPKT HCM, 2012.
6. Peter J.Ashenden, The VHDL cookbook, the first edition, july 1990

Website:

http://www.datasheet.com

http://www.digilentinc.com

http://www.gmvhdl.com/VHDL.html

http://hdllib.at.infoseek.co.jp/hdl/vhdl/index.html

http://www.soccentral.com

http://scripturn.com/tag_vhdl.html

http://www.cs.ucr.edu/~vahid/

http://www.cs.ucr.edu/~vahid/courses/122a_f99/zcookbook/index.html

http://www.echelonembedded.com

http://www.altera.com/

http://e-www.motorola.com/index.html

http://www.xilinx.com/

http://www.netrino.com/Publications/Glossary/

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT THÀNH PHỐ HỒ CHÍ MINH

www.hcmute.edu.vn