

Đại học Quốc gia Thành phố Hồ Chí Minh
Trường Đại học Bách Khoa
Khoa Khoa học và Kỹ thuật Máy tính



Luận Văn Tốt Nghiệp

**Đề tài: Nhận dạng biển số xe trên
nền tảng hệ thống vi mạch không
đồng nhất**

Hội đồng : Kỹ Thuật Máy Tính
Giáo viên hướng dẫn : TS. Nguyễn Trần Hữu Nguyên
Giáo viên phản biện : TS. Phạm Quốc Cường

Sinh viên thực hiện : Nguyễn Xuân Nam - 1512098

Ngày 19 tháng 5 năm 2019

I think, therefore I am;
— René Descartes

Lời cam đoan

Nhóm cam đoan mọi điều được ghi trong báo cáo, cũng như mã nguồn là do nhóm tự thực hiện - trừ các kiến thức tham khảo có trích dẫn cũng như mã nguồn mẫu do chính nhà sản xuất cung cấp, hoàn toàn không sao chép từ bất cứ nguồn nào khác. Nếu lời cam đoan trái với sự thật, nhóm xin chịu mọi trách nhiệm trước Khoa và Nhà trường.

Nhóm sinh viên thực hiện đề tài

Lời cảm ơn

Sau bốn năm theo học tại khoa Khoa học Kỹ thuật Máy tính trường Đại học Bách khoa TP. Hồ Chí Minh, em đã được áp dụng tất cả những kiến thức đã học để thực hiện luận văn tốt nghiệp, đề tài cuối cùng mang tính quyết định kết quả học tập và quá trình tốt nghiệp.

Em xin chân thành cảm ơn quý thầy cô trong Khoa đã mang đến những kiến thức bổ ích trong suốt quãng đời sinh viên, đặc biệt là thầy Phạm Hoàng Anh đã giúp đỡ em trong quá trình thực hiện cũng như quá trình hoàn thiện luận văn. Em cũng gửi lời cảm ơn đến một số bạn bè trong Khoa đã hỗ trợ giúp em tiết kiệm được thời gian tìm hiểu đề tài.

Em đã cố gắng hết sức để tránh các sai sót, nhưng nếu có phát hiện thì mong quý thầy cô góp ý để em càng hoàn thiện đề tài hơn. Cuối cùng em xin gửi lời chúc sức khỏe và cảm ơn chân thành nhất.

Nhóm sinh viên thực hiện đề tài



Tóm tắt

Mục lục

Lời cam đoan

Lời cảm ơn i

Tóm tắt ii

Mục Lục Hình v

Danh Sách Bảng v

Thuật ngữ & từ viết tắt vii

1 Giới thiệu đề tài 1

1.1 Nhiệm vụ cần đạt 1

1.2 Sơ đồ khối 1

1.3 Hoạt động 1

2 Các công trình liên quan 2

2.1 BNN vs CNN 2

3 Cơ sở lý thuyết 3

3.1 Vivado HLS 3

3.2 Vivado 3

3.3 Bo mạch Ultra96 3

3.4 PYNQ Framework 3

3.4.1 Pynq là gì? 3

3.4.2 PYNQ Overlays 5

3.4.3 Phương pháp thiết kế overlay 6

4	Quá trình hiện thực	8
4.1	Các giai đoạn	8
4.2	LP Detection	8
4.3	Character Segmentation	8
4.4	Character Regcongition	8
5	Kịch bản thử nghiệm	9
5.1	Mục tiêu của thử nghiệm	9
5.2	Quá trình thử nghiệm	9
5.2.1	Kịch bản I:	9
6	Kết luận	10
6.1	Đánh giá kết quả	10
6.1.1	Thành quả đạt được	10
6.1.2	Một số hạn chế	10
6.2	Hướng phát triển	10
A	Hướng dẫn thực thi ứng dụng	11
	Tài liệu tham khảo	12

adds an entry to the table of contents

Danh sách hình vẽ

3.1	Sơ đồ khối của Pynq	4
3.2	Sơ đồ khối của Zynq	5

Thuật ngữ & từ viết tắt

API Application Programming Interface

ASIC Application Specific Integrated Circuit

GPIO General Purpose Input Output

HLS High-Level Synthesis

IP Intellectual Property

IPython Interactive Python

PFGA Field Programmable Gate Array

PL Programmable Logic

PS Processing System

PYNQ Python Productivity for Zynq

..... SoC Systems on Chip

1 Giới thiệu đề tài

1.1 Nhiệm vụ cần đạt

1.2 Sơ đồ khối

1.3 Hoạt động

2 Các công trình liên quan

2.1 BNN vs CNN

3 Cơ sở lý thuyết

3.1 Vivado HLS

3.2 Vivado

3.3 Bo mạch Ultra96

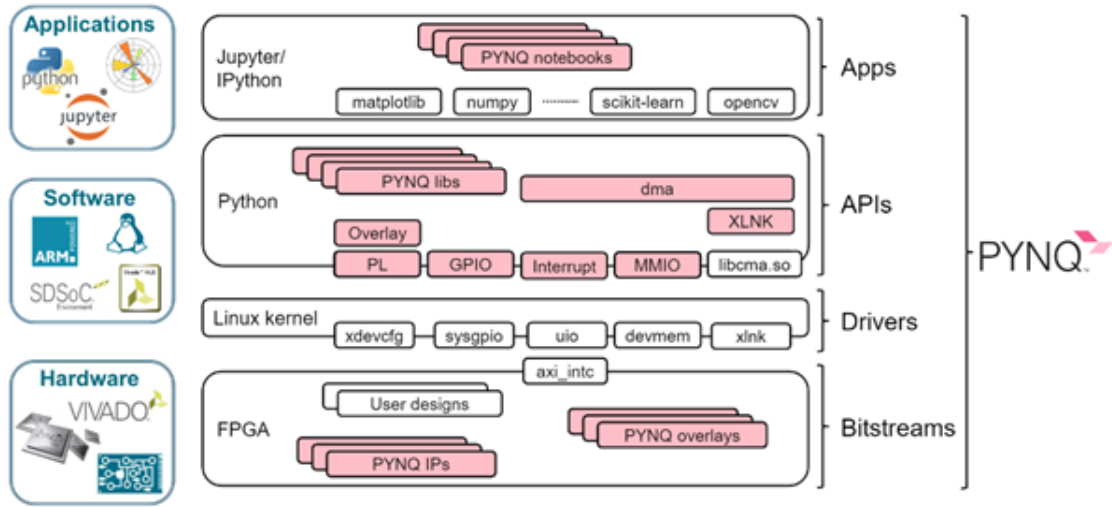
3.4 PYNQ Framework

3.4.1 Pynq là gì?

PYNQ, Python Productivity for Zynq, là một dự án mã nguồn mở của Xilinx giúp dễ dàng thiết kế những hệ thống nhúng với Xilinx Zynq[1] Systems on Chips (SoCs). Bằng việc sử dụng ngôn ngữ Python và các thư viện, người thiết kế có thể khai thác những thế mạnh của FPGA (hay programmable logic) cũng như của bộ vi xử lý trên Zynq xây dựng những hệ thống nhúng thú vị và hiệu quả. Đặc biệt, Pynq cho phép những kiến trúc sư, những kỹ sư và những người lập trình về hệ thống nhúng có thể sử dụng thiết bị Zynq mà không phải sử dụng đến các công cụ thiết kế ASIC-style.

Pynq đạt được điều đó bởi:

1. Những mạch lô-gíc lập trình được xem như những thư viện phần cứng hay còn gọi là overlays. Những overlays này tương tự như những thư viện phần



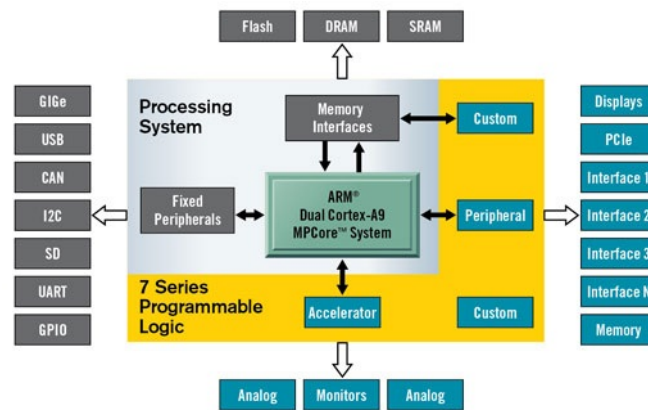
Hình 3.1: Sơ đồ khối của Pynq

mềm. Một kỹ sư phần mềm có thể lựa chọn những overlay phù hợp nhất với ứng dụng của mình. Một overlay có thể được truy cập thông qua giao diện ứng dụng lập trình (API). Việc tạo một overlay mới đòi hỏi người kỹ sư có kiến thức về thiết kế mạch lô-gíc lập trình.

2. Pynq sử dụng Python cho việc lập trình trên cả những vi xử lý và những overlay. Mà Python là một ngôn ngữ bậc cao rất hiệu quả. Ngày nay, C hay C++ là những ngôn ngữ lập trình cho hệ thống nhúng phổ biến nhất. Nhưng, Python là ngôn ngữ có mức độ trừu tượng tốt hơn và mang lại hiệu suất cao hơn cho người lập trình. Tuy nhiên không có sự lựa chọn nào là tốt tuyệt đối. Pynq sử dụng CPython - được viết bằng C, và tích hợp với hàng ngàn thư viện C. Nếu hướng tới tăng hiệu suất lập trình thì nên sử dụng môi trường Python và nếu muốn tăng hiệu năng thì lập trình C nên được ưu tiên.
3. Pynq là dự án mã nguồn mở hướng tới có thể hoạt động trên bất kỳ nền tảng hay hệ điều hành nào. Đạt được điều này bằng việc thông qua kiến trúc trên nền web, còn được gọi là browser agnostic. Nó tích hợp nền tảng mã nguồn mở Jupyter notebook để chạy một nhân Interactive Python (IPython) và một web server trực tiếp trên vi xử lý ARM của thiết bị Zynq.

3.4.2 PYNQ Overlays

Thiết bị Zynq là một hệ thống trên chip gồm hai phần cơ bản là bộ vi xử lý (gọi là Processing System hay PS) và tích hợp với phần FPGA (còn gọi là Programmable Logic hay PL). Phần PS bao gồm hàng tá những thiết bị ngoại vi (memory controllers, USB, Uart, IIC, SPI,...) và có thể mở rộng ra những IP phần cứng bổ sung trên một overlay PL (xem hình 3.2).



Hình 3.2: Sơ đồ khối của Zynq

Overlay, hay còn gọi là thư viện phần cứng, là những thiết kế FPGA. Nó mở rộng và kết nối phần PS của Zynq với phần PL. Overlay có thể được sử dụng để gia tốc một ứng dụng phần mềm hay tùy chỉnh nền tảng phần cứng cho phù hợp với những ứng dụng đặc thù.

Ví dụ, xử lý ảnh là một ứng dụng điển hình, nơi mà FPGA có thể được sử dụng để gia tốc. Một người lập trình phần mềm có thể sử dụng một overlay tương tự như sử dụng thư viện phần mềm để chạy một số hàm xử lý ảnh trên nền tảng FPGA. Những overlay có thể được tải lên FPGA một cách linh hoạt khi được yêu cầu, giống như một thư viện phần mềm. Những hàm xử lý ảnh riêng biệt có thể được hiện thực bằng những overlay khác nhau và được tải lên dựa vào yêu cầu từ Python.

Pynq cung cấp một giao diện Python cho phép Python chạy trên phần PS có thể điều khiển những overlay chạy trên phần PL. Thiết kế FPGA là công việc đặc thù mà đòi hỏi sự hiểu biết của người kỹ sư về phần cứng. Những Pynq overlay có được tạo ra bởi người thiết kế phần cứng và được gói lại với Pynq Python API. Người phát triển phần mềm có thể sử dụng giao diện Python để lập trình và điều khiển

những overlay phần cứng mà không cần tự tay thiết kế chúng. Điều này tương tự như sử dụng thư viện phần mềm được tạo ra bởi những người phát triển nhiều kinh nghiệm.

3.4.3 Phương pháp thiết kế overlay

Đã được nêu lên trong phần 3.4.2, overlay tương tự như những thư viện phần mềm. Một người lập trình có thể tải overlay vào phần PL để cung cấp những chức năng mà ứng dụng cần.

Một overlay là một lớp những thiết kế PL. Những thiết kế PL này luôn được tối ưu hóa cho một công việc nhất định. Tuy nhiên cũng có những overlay được thiết kế để có thể tùy chỉnh và sử dụng lại cho những tập ứng dụng nhất định. Một Pynq overlay có giao diện Python. Điều này cho phép người lập trình có thể sử dụng nó như một gói Python.

Người lập trình phần mềm có thể sử dụng overlay mà không tạo ra chúng vì điều này cần hiểu biết về thiết kế phần cứng. Tuy nhiên có một số công việc cần tới quá trình tạo một overlay:

- Tích hợp Python và C.
- Tạo API cho Python overlay.
- Thiết lập board
- Tạo gói Python

Phần này sẽ chỉ ra quá trình để tạo một overlay và tích hợp nó vào Pynq.

Thiết kế overlay

Một overlay bao gồm hai thành phần; phần thiết kế PL (bitstream) và phần sơ đồ khối (tệp tcl). Thiết kế overlay là công việc đặc thù của kỹ sư phần cứng. Phần này đòi hỏi đến kiến thức về xây dựng thể thống Zynq cùng với các công cụ thiết kế Vivado.

Thiết kế PL

Phần mềm Xilinx Vivado được sử dụng cho việc tạo một thiết kế Zynq. Một bitstream hay một tệp nhị phân sẽ được sinh ra và được sử dụng cho lập trình trên Zynq PL. Những thiết kế phần cứng hỗ trợ lập trình trong những IP (Intellectual property), được sử dụng trong những Pynq overlay. Khi IP được tạo ra, thiết kế PL sẽ được thực hiện giống như bất kỳ thiết kế Zynq nào. IP trong overlay, được điều khiển bởi Pynq, sẽ được ánh xạ vùng nhớ và gắn với GPIO. Pynq cung cấp thư viện để tương tác với thiết kế PL và được sử dụng để tạo driver.

Tệp overlay Tcl

Tệp Tcl từ Vivado IP được sử dụng bởi Pynq để tự động xác định tùy chỉnh hệ thống Zynq. IP bao gồm phiên bản, ngắt, reset, và các tín hiệu khác. Dựa trên những thông tin này, một phần của việc tùy chỉnh hệ thống được tự động tinh chỉnh từ Pynq, những driver có thể tự động được gán, một số đặc điểm có thể được kích hoạt hay gỡ bỏ và những tín hiệu có thể được kết nối đến các phương thức trả lời của Python.

Tệp Tcl phải được sinh ra và cung cấp cùng với tệp nhị phân (bitstream) như là một phần của overlay. Tệp Tcl có thể được sinh ra trong công cụ Vivado bằng cách xuất sơ đồ khối tích hợp IP (IP Integrator block diagram) tại cuối quá trình thiết kế overlay. Tệp Tcl phải được cung cấp cùng với tệp nhị phân khi tải overlay. Lớp Pynq PL sẽ tự động phân tích tệp Tcl.

Tên của tệp Tcl cần được giống với tên của tệp nhị phân. Ví dụ my_overlay.bit và my_overlay.tcl.

4 Quá trình hiện thực

4.1 Các giai đoạn

Lấy biển số từ hình

Cắt các ký tự từ biển số

Xác định các ký tự.

4.2 LP Detection

4.3 Character Segmentation

4.4 Character Recognition

5 Kịch bản thử nghiệm

5.1 Mục tiêu của thử nghiệm

5.2 Quá trình thử nghiệm

5.2.1 Kịch bản I:

6 Kết luận

6.1 Đánh giá kết quả

6.1.1 Thành quả đạt được

6.1.2 Một số hạn chế

6.2 Hướng phát triển

A Hướng dẫn thực thi ứng dụng

Tài liệu tham khảo

[1] Xilinx Zynq là gì?

<https://www.xilinx.com/products/silicon-devices/soc/zynq-7000.html>