

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



**LUẬN VĂN THẠC SĨ
VÕ VĂN TRUNG**

**NGHIÊN CỨU FPGA TRONG THIẾT KẾ
CPU 32 BIT THEO KIẾN TRÚC MIPS**

NGÀNH: KỸ THUẬT ĐIỆN TỬ-60520203



Tp. Hồ Chí Minh, tháng 10/2015

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

LUẬN VĂN THẠC SĨ
VÕ VĂN TRUNG

NGHIÊN CỨU FPGA TRONG THIẾT KẾ CPU 32 BIT THEO
KIẾN TRÚC MIPS

NGÀNH: KỸ THUẬT ĐIỆN TỬ-60520203

Hướng dẫn khoa học:

TS . NGUYỄN MINH NGỌC

Tp.Hồ Chí Minh,tháng 10/2015

MỤC LỤC

Chương 1_TỔNG QUAN.....	7
1.1 Tổng quan chung về lĩnh vực nghiên cứu, các kết quả nghiên cứu trong và ngoài nước đã công bố	7
1.1.1 Tình hình ngoài nước.....	7
1.1.2 Tình hình trong nước	10
1.2 Mục đích của đề tài.....	11
1.3 Nhiệm vụ của đề tài và giới hạn đề tài	12
1.4 Phương pháp nghiên cứu	13
Chương 2_CƠ SỞ LÝ THUYẾT	14
2.1. Giới thiệu về ngôn ngữ Verilog.....	14
2.1.1. Mục đích sử dụng verilog	14
2.1.2. Một số quy ước thiết kế	15
2.2. Công nghệ FPGA.....	16
2.2.1. Công nghệ FPGA là gì.....	16
2.2.2. Tại sao công nghệ FPGA được chú ý	16
2.2.3. Kiến trúc chung của một FPGA	17
2.2.4. So sánh CPLD và FPGA.....	17
2.3. Kiến trúc của MIPS	18
2.3.1. Giới thiệu.....	18
2.3.2. Tập lệnh MIPS (một phần).....	19
2.3.2.1. Các lệnh loại R	23
2.3.2.2. Các lệnh loại I.....	24
2.3.2.3. Các lệnh loại J.....	24
2.3.3. Thanh ghi MIPS.....	25
2.4. Kỹ thuật thiết kế đường ống (pipeline).....	27
2.4.1. Giới thiệu.....	27
2.4.2. Thiết kế đường ống.....	30
2.5. Lập trình, trình biên dịch và mô phỏng vi xử lý MIPS	31
2.5.1. Giao diện PCSpim	31
2.5.2. Mở một chương trình MIPS với PCSpim	33
2.5.3. Tập tin mã lệnh chương trình mẫu	33
2.5.4. Các chế độ chạy mô phỏng chương trình	34

2.5.4.1. Chạy đơn giản toàn bộ chương trình	34
2.5.4.2. Chạy từng dòng lệnh	35
2.5.4.3. Chạy nhiều dòng lệnh	37
Chương 3_THIẾT KẾ CPU 32 BIT	38
3.1. Các vấn đề cần giải quyết	38
3.2. Phương pháp giải quyết	38
3.2.1. Đường dữ liệu đơn chu kỳ	39
3.2.1.1. Đơn vị điều khiển ALU	44
3.2.1.2. Đơn vị điều khiển chính	47
3.2.2. Đường dữ liệu đường ống	48
3.3. Các vấn đề đã giải quyết và kết quả đạt được	52
Chương 4_THIẾT KẾ NGOẠI VI TRÊN HỆ THỐNG KIT FPGA	57
4.1 Phần cứng trên FPGA	57
4.2 Giải mã địa chỉ lựa chọn ngoại vi	58
4.3. Uart-Rs232	58
4.3.1. Giới thiệu	58
4.3.2. Hoạt động Uart	60
4.4. Timer	61
4.4.1. Thanh ghi điều khiển (control)	62
4.4.2. Thanh ghi trạng thái (status)	62
4.4.3. Chu kỳ đếm	63
4.4.4. Giá trị của thanh ghi counter	63
4.5. GPIO	63
4.6. Thiết kế các ứng dụng mô tả chức năng các ngoại vi	63
4.6.1. Quá trình kiểm tra chung	64
4.6.2. Giải thuật điều khiển Timer	65
4.6.3. Giải thuật điều khiển giao tiếp Uart	65
4.6.4. Giải thuật điều khiển dịch led	66
Chương 5_KIỂM TRA MÔ PHỎNG VÀ THỰC NGHIỆM TRÊN KIT FPGA	67
5.1. Mô phỏng nhóm lệnh R	68
5.2. Mô phỏng nhóm lệnh I	70
5.3. Mô phỏng nhóm lệnh loại J	71
5.7. Kết quả thực nghiệm trên kit FPGA Alteral DE2	74
5.7.1. Giới thiệu kit Alteral DE2	74

5.7.2. Timer	75
5.7.3. Giao tiếp Uart	75
5.7.4. Điều khiển led.....	76
Chương 6_KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN ĐỀ TÀI.....	77
6.1. Kết luận	77
6.2. Hướng phát triển đề tài	77
Tài liệu tham khảo	79

MỤC LỤC BẢNG

Bảng 2 - 1: Một phần tập lệnh Mips.....	20
Bảng 2 - 2: Một số lệnh loại R.....	23
Bảng 2 - 3: Một số lệnh loại I	24
Bảng 2 - 4: Một số lệnh loại J	25
Bảng 2 - 5: Thanh ghi MIPS	26
Bảng 3 - 1: Giá trị tín hiệu điều khiển cho ALU	44
Bảng 3 - 2: Tín hiệu điều khiển ALUcontrol.....	45
Bảng 3 - 3: Giá trị các tín hiệu vào ra khỏi ALUcontrol.....	46
Bảng 3 - 4: Giá trị các tín hiệu điều khiển cho các lệnh	47
Bảng 4-1: Tập thanh ghi của Uart.....	60
Bảng 4-2: Chức năng của thanh ghi control	62
Bảng 4-3: Chức năng của thanh ghi status	62

MỤC LỤC HÌNH VẼ

Hình 2 - 1: Cấu trúc tổng thể của một FPGA	17
Hình 2 - 2: Đường dữ liệu đa chu kỳ.....	28
Hình 2 - 3: Xung clock đa chu kỳ	29
Hình 2 - 4: Hoạt động theo kỹ thuật đường ống	30
Hình 2 - 5: Hoạt động đường ống cho nhóm lệnh loại R.....	31
Hình 2 - 6: Giao diện người dùng của PCSpim	32
Hình 2 - 7: Mở tập tin trong PCSpim.....	33
Hình 2 - 8: Cửa sổ Console.....	34
Hình 2 - 9: Chọn địa chỉ bắt đầu cho PCSpim.....	35
Hình 2 - 10: Chèn breakpoint để dừng trình mô phỏng	35
Hình 2 - 11: Điểm chèn Breakpoint	36
Hình 2 - 12: Hộp thoại làm việc với Breakpoint	36
Hình 2 - 13: Hộp thoại hỏi số bước chạy	37
Hình 3 - 1: Đường dữ liệu đơn chu kỳ	39
Hình 3 - 2: Giai đoạn nạp mã lệnh	40
Hình 3 - 3: Giải mã lệnh	41
Hình 3 - 4: Thực thi lệnh loại R.....	42
Hình 3 - 5: Thực thi lệnh Load/Store	43
Hình 3 - 6: Thực thi lệnh nhảy hoặc rẽ nhánh	43
Hình 3 - 7: Mạch điều khiển các tín hiệu cho ALUcontrol.....	46
Hình 3 - 8: Mạch tổ hợp của khối điều khiển chính.....	48
Hình 3 - 9: Chia tầng đường ống cho đường dữ liệu	49

Hình 3 - 10: Ưu điểm của kỹ thuật đường ống	51
Hình 3 - 11: Các khối chức năng của CPU thiết kế	52
Hình 4 -1: Mô hình hệ thống thiết kế trên FPGA	57
Hình 4 – 2: Quá trình giải mã địa chỉ để lựa chọn ngoại vi	58
Hình 4-3: Khung truyền dữ liệu	59
Hình 4-4: Chu kỳ bit và thời gian lấy mẫu	59
Hình 4-5: Thành phần chính của bộ Uart	60
Hình 4-7: Sơ đồ điều khiển GPIO	63
Hình 4 – 8: Lưu đồ kiểm tra chung	64
Hình 4-9: Lưu đồ điều khiển đếm timer	65
Hình 4 -10: Lưu đồ điều khiển giao tiếp với Uart	65
Hình 4-11: Lưu đồ dịch led.....	66
Hình 5 - 1: Giao diện với PCSpim thực hiện lệnh loại R.....	69
Hình 5 - 2: Kết quả mô phỏng trên ModelSim nhóm lệnh loại R	69
Hình 5 - 3: Giao diện PCSpim thực hiện lệnh loại I	70
Hình 5 - 4: Mô phỏng ModelSim với nhóm lệnh loại I	71
Hình 5 - 5: Giao diện PCSpim thực hiện lệnh loại J	72
Hình 5 - 6: Mô phỏng ModelSim với nhóm lệnh loại J	73
Hình 5 – 7: Sơ đồ kit FPGA Alteral DE2	74
Hình 5 - 8: kết quả thực nghiệm cho bộ Timer đếm	75
Hình 5 - 9: Kết quả thực nghiệm với việc cộng trừ 2 số giao tiếp Uart.....	75
Hình 5 - 10: Kết quả dịch led trên kit.....	76

Chương 1

TỔNG QUAN

1.1 Tổng quan chung về lĩnh vực nghiên cứu, các kết quả nghiên cứu trong và ngoài nước đã công bố

1.1.1 Tình hình ngoài nước

Vi xử lý là một mạch điện tích hợp bao gồm hầu hết các chức năng của một đơn vị xử lý trung tâm của máy tính. Chip vi xử lý đầu tiên xuất hiện vào những năm đầu của thập niên 1970 và được sử dụng cho các bộ tính toán điện tử, sử dụng thuật toán trên số BCD (binary-coded decimal). Ngay sau đó, các hệ thống sử dụng vi xử lý 4-bit và 8-bit, như là thiết bị đầu cuối, máy in, các thiết bị tự động,... lần lượt ra đời. Những vi xử lý 8-bit với địa chỉ 16-bit bắt đầu tạo ra những máy tính nhỏ dùng cho mục đích chung đầu tiên, từ những năm giữa của thập niên 1970.

Trong suốt thập niên 1960, vi xử lý thường được xây dựng từ vài chục đến vài trăm transistor. Việc tích hợp toàn bộ vi xử lý vào trong một con chip làm giảm chi phí xử lý rất lớn. Từ khởi đầu khiêm tốn đó, những cải tiến liên tục tiếp theo sau đó về khả năng của vi xử lý đã thay thế những vi xử lý mà ngày nay hầu hết đã lỗi thời bằng những vi xử lý cực mạnh, sử dụng trong mọi thứ từ hệ thống nhúng, thiết bị cầm tay đến những siêu máy tính.

Từ những năm đầu của thập niên 1970, sự tăng trưởng về khả năng của vi xử lý đã tuân theo luật Moore, nghĩa là số lượng transistor tích hợp bên trong một chip đơn sẽ tăng gấp đôi sau mỗi 2 năm. Mặc dầu, những tính toán ban đầu cho rằng mức độ tích hợp sẽ tăng gấp đôi theo từng năm. Sau đó, Moore đã điều chỉnh chu kỳ tăng trưởng này lại thành 2 năm.

Thiết kế 32-bit quan trọng nhất là chip MC68000 (hay còn được gọi là chip 68K) được giới thiệu vào năm 1979. Chip 68K có những thanh ghi 32-bit nhưng sử dụng đường dữ liệu nội 16-bit và đường dữ liệu bên ngoài 16-bit, và chỉ hỗ trợ 24-bit địa chỉ. Motorola đã mô tả nó như là vi xử lý 16-bit, mặc dù kiến trúc của nó là 32-bit. Sự kết hợp của vùng không gian bộ nhớ lớn, tốc độ cao và giá thành thấp làm nó trở thành thiết kế CPU nổi tiếng nhất.

Vi xử lý 32-bit đầu tiên trên thế giới bao gồm đầy đủ luồng dữ liệu 32-bit, cấu trúc bus 32-bit, và địa chỉ 32-bit, là chip BELLMAC-32A của hãng AT&T BELL. Mẫu đầu tiên của dòng chip này ra đời vào năm 1980, sau đó sản xuất hàng loạt vào năm 1982. Sau khi AT&T BELL bị tách ra vào năm 1984, chip BELLMAC-32A được đổi tên thành WE32000, và có 2 phiên bản tiếp theo là WE32100 và WE32200. Những vi xử lý này được sử dụng trong máy tính thu nhỏ AT&T 3B5 và 3B15; trong máy tính để bàn đầu tiên của thế giới 3B2; trong máy tính Laptop đầu tiên của thế giới Companion; trong máy tính có kích cỡ nhỏ bằng cuốn sách (book-sized) đầu tiên của thế giới Alexander. Tất cả những máy tính này đều chạy hệ điều hành UNIX System V.

VXL 32-bit đầu tiên của Intel là iAPX 432, được giới thiệu đầu tiên vào năm 1981, nhưng không thành công về mặt thương mại. Nó có kiến trúc Hướng đối tượng dựa trên khả năng (capability-based object-oriented), nhưng khả năng thực hiện (performance) thấp hơn so với kiến trúc khác của chính Intel tại thời điểm đó là 80286 (được giới thiệu vào năm 1982). Intel 80286 chạy nhanh hơn gấp 4 lần so với iAPX 432 trên cùng các chương trình ứng dụng kiểm tra. Hậu quả của iAPX432 một phần là do dự án bị thúc ép về mặt thời gian, và do đó trình biên dịch được sử dụng là Ada chưa được tối ưu.

Chip vi xử lý ARM xuất hiện đầu tiên năm 1985. Đây là vi xử lý 32-bit có cấu trúc kiểu RISC. ARM đã trở nên vượt trội trong lĩnh vực vi xử lý dùng cho hệ thống

nhúng (embedded system), với ưu điểm là hiệu suất cao, mô hình bản quyền, và có nhiều lựa chọn cho việc sử dụng công cụ phát triển hệ thống. Những nhà sản xuất bán dẫn thường mua các lõi VXL như ARM11 và tích hợp chúng vào trong sản phẩm riêng của họ (system on a chip). Rất ít nhà sản xuất được cấp quyền để sửa đổi kiến trúc lõi vi xử lý ARM. Hầu hết điện thoại di động đều có một vi xử lý ARM. VXL ARM cũng xuất hiện nhiều trong các sản phẩm điện tử khác.

Cũng trong thời gian này (những năm đầu thập niên 1980), hãng National Semiconductor (NS) đã giới thiệu vi xử lý 32-bit rất đơn giản với 16 chân ngõ ra, được gọi là NS16032. Phiên bản đầy đủ tiếp theo là NS32032. Sau đó, phiên bản tiếp theo được giới thiệu là NS32132, với 2 lõi CPU sử dụng chung bus bộ nhớ, cùng được thực hiện bên trong một chip đơn. Chip NS32032 chạy nhanh hơn MC68010, nhưng NS32332 lại hoạt động chậm hơn MC68020 (chip của 2 hãng được giới thiệu xấp xỉ cùng thời điểm). Thế hệ chip thứ 3 của NS là NS32532 hoạt động nhanh gấp đôi chip MC68030 (2 chip này được giới thiệu cùng thời điểm). Việc xuất hiện của các vi xử lý AM29000 và MC88000 (mà ngày này không còn nữa) ảnh hưởng đến kiến trúc vi xử lý cuối cùng của hãng NS là NS32764. Kiến trúc được cải tiến, sử dụng kiến trúc superscalar RISC, bus 64-bit, nó có khả năng xử lý 32000 lệnh liên tiếp theo thời gian thực. Khi hãng National Semiconductor quyết định rút khỏi thị trường hệ điều hành UNIX, dòng VXL này được thiết kế lại thành vi xử lý Swordfish Embedded với một tập hợp nhiều ngoại vi bên trong chip. Chip VXL này được tung ra vào thị trường máy in laze với giá thành quá đắt và bị đào thải. Nhóm thiết kế đó qua Intel và ở đó thiết kế VXL Pentium, với kiến trúc rất giống với kiến trúc của NS32764.

Vi xử lý MIPS R2000 (1984) và MIPS R3000 (1989) rất thành công trong lĩnh vực VXL RISC 32-bit. Chúng được sử dụng cho máy tính để bàn và máy chủ bởi công ty SGI.

Những thiết kế khác phải kể đến Zilog Z80000, xuất hiện để thâm nhập thị trường quá trễ và nhanh chóng biến mất.

Vào cuối thập niên 1980, “cuộc chiến giữa những dòng vi xử lý” bắt đầu làm biến mất một số dòng vi xử lý. Từ 1985 đến nay, kiến trúc VXL 32-bit x86 trở nên chiếm lĩnh trong thị trường máy tính để bàn, laptop, server. Dòng VXL này ngày càng chạy nhanh hơn với nhiều tính năng hơn. Intel đã bán kiến trúc những phiên bản vi xử đầu tiên cho những công ty khác, nhưng từ chối bán kiến trúc của Pentium. Vì thế, các hãng như AMD và Cyrix sau đó đã xây dựng những kiến trúc VXL 32-bit của riêng họ. Trong suốt thời kỳ này, những vi xử lý này đã không ngừng cải tiến về độ tích hợp và khả năng hoạt động (instructions/second). Bộ vi xử lý Pentium của Intel có lẽ là dòng VXL 32-bit nổi tiếng nhất.

Cũng trong giai đoạn này (1985 đến nay), kiến trúc VXL ARM cũng chiếm lĩnh thị trường VXL 32-bit trong lĩnh vực hệ thống nhúng (embedded system). Hầu như mỗi điện thoại di động ngày nay đều có một VXL ARM bên trong. VXL ARM cũng rất thông dụng trong các ứng dụng nhúng khác.

Xu hướng phát triển của VXL 32-bit ngày nay cạnh tranh nhau ở tốc độ thực hiện (tần số + hiệu suất hoạt động) và công suất tiêu thụ. Đôi khi, hai yêu cầu này không xảy ra đồng thời. Đối với những hệ thống cần tốc độ xử lý nhanh như máy tính, server, máy chơi game, các ứng dụng thời gian thực... thì tốc độ thực hiện là ưu tiên hàng đầu. Đối với những thiết bị cầm tay, thì công suất tiêu thụ là yêu cầu trước tiên. Tuy nhiên, trong nhiều thiết bị vừa cần tốc độ xử lý nhanh lại vừa hạn chế về mặt năng lượng (VD: điện thoại di động), cả 2 yếu tố tốc độ và công suất đều rất cần thiết.

1.1.2 Tình hình trong nước

Ngành công nghiệp điện tử Việt Nam vốn rất nhỏ bé so với các nước trên thế giới, các sản phẩm điện tử mới chỉ dừng lại ở công việc lắp ráp mà người ta quen gọi là

công nghệ Tuốc-nơ-vít. Sản xuất trong nước chủ yếu là nhập linh kiện, công nghiệp phụ trợ kém phát triển, có chăng thì sản xuất các sản phẩm đơn giản, hàm lượng chất xám tích lũy nghèo nàn, giá trị gia tăng thấp. Ngành thiết kế vi mạch chỉ mới bắt đầu hình thành trong 3 năm trở lại đây (từ năm 2007 đến nay) với sự ra đời của Trung tâm Đào tạo và Thiết kế Vi Mạch (ICDREC). Trong đó, ICDREC đã tập hợp và xây dựng đội ngũ thiết kế vi mạch mạnh với những thành công ban đầu. Cụ thể là sản xuất thành công các thế hệ chip đầu tiên như: Chip VXL 8-bit SigmaK3 (Chip digital đầu tiên của Việt Nam), Chip VXL 8-bit VN8-01, Chip LDO TH7150 (chip analog đầu tiên của Việt Nam),...

Chip VXL 32-bit đầu tiên của Việt Nam là chip VXL VN1632, do ICDREC thiết kế (được công bố vào tháng 10/2010). Chip VN1632 được thiết kế với công nghệ 130nm, tần số hoạt động tối đa 100MHz. Về tính năng, VN1632 đảm bảo hầu hết các tính năng của một VXL 32-bit thông thường. Về chất lượng thiết kế (tần số hoạt động, hiệu suất thực hiện, công suất tiêu thụ, công nghệ thiết kế,...), với trình độ còn hạn chế của đội ngũ thiết kế, chip VN1632 chưa thể so sánh với các chip VXL 32-bit hiện đại trên thế giới hiện nay.

1.2 Mục đích của đề tài

- Tìm hiểu và nắm bắt được ngôn ngữ lập trình mô tả phần cứng Verilog.
- Tìm hiểu kiến trúc tập lệnh của Mips.
- Tìm hiểu về FPGA
- Thực hiện thiết kế phần cứng cho vi xử lý RISC theo kiến trúc tập lệnh của MIPS.
- Toàn bộ thiết kế có số lượng cổng logic đủ lớn để nạp và chạy thực tế trên các kit thực hành FPGA.
- Nắm vững kỹ thuật thiết kế đường ống để xây dựng phần cứng xử lý phù hợp cho các lệnh hỗ trợ.

1.3 Nhiệm vụ của đề tài và giới hạn đề tài

❖ Nhiệm vụ của đề tài

- Qua tìm hiểu về FPGA trong lĩnh vực thiết kế CPU, ta tiến hành thiết kế CPU 32 bit, theo kiến trúc tập lệnh MIPS bằng ngôn ngữ verilog viết trên phần mềm Quartus II của Altera.
- CPU thiết kế trên cơ chế đường ống (pipeline). Gồm có 5 tầng IF, ID, EX, MEM, WB.
- Bộ nhớ ROM 1K gồm 256 lệnh
- Bộ nhớ dữ liệu 4K
- Tần số 2MHz
- Có 4 port điều khiển
- Viết các module cho các tầng theo kiến trúc Mips.
- Viết chương trình Test Bench để kiểm tra.
- Tiến hành mô phỏng các nhóm lệnh R, I, J trên phần mềm Modelsim và đánh giá với quá trình thực thi lệnh ở phần mềm lập trình cho vi xử lý MIPS PCSpim.
- Thiết kế ngoại vi GPIO, TIMER, UART và tiến hành kiểm tra các chương trình trên kit FPGA Altera DE2-115

❖ Giới hạn của đề tài

Một điểm còn thiếu của đề tài là chưa hỗ trợ xử lý ngắt cho vi xử lý, chương trình do đó chỉ được đọc trực tiếp từ bộ nhớ ROM và thực thi theo các mã lệnh đã lập trình.

1.4 Phương pháp nghiên cứu

- Do đây là một đề tài cũng khá mới mẻ với bản thân, bước đầu nghiên cứu và tìm hiểu về lĩnh vực thiết kế Chip. Việc thực hiện được tiến hành một cách tuần tự từ đơn giản cho đến phức tạp, từ mức sơ đồ khối lớn đến các khối con. Trình tự sẽ như sau:

+ Phân tích hoạt động của vi xử lý bắt đầu với từng nhóm lệnh riêng lẻ. Ví dụ như: nhóm R, nhóm I, nhóm J sẽ chọn ra 1 lệnh để thực hiện và làm dần cho các lệnh khác đến khi xây dựng xong chức năng cho tất cả các lệnh.

+ Ghép các đường dữ liệu của từng nhóm lệnh lại với nhau, nếu các đường tín hiệu xảy ra cùng đưa đến cho 1 đầu vào thì cần chèn thêm các mạch chọn lựa đầu vào trong các trường hợp này. Kết quả là đường dữ liệu cho tất cả các lệnh hoạt động trong 1 chu kỳ xung clock.

+ Chia đường dữ liệu đơn chu kỳ thành các tầng đường ống dữ liệu.

+ Thay đổi đường dữ liệu để có thể xử lý nhiều lệnh cùng lúc, khắc phục các vấn đề của đường dữ liệu khi hoạt động theo đường ống.

Kết luận chương 1

Trong chương này tác giả đã đánh giá được tổng quan về tình hình phát triển chung của các dòng vi xử lý xưa và nay, đồng thời đã chỉ rõ được mục đích, phạm vi nghiên cứu, phương pháp nghiên cứu của đề tài được giao. Từ đó có định hướng để làm các tiếp các chương tiếp theo được dễ dàng.

Chương 2

CƠ SỞ LÝ THUYẾT

2.1. Giới thiệu về ngôn ngữ Verilog

2.1.1. Mục đích sử dụng verilog

Hệ thống số là một hệ thống phức tạp bậc cao. Ở cấp độ chi tiết nhất, chúng có thể bao gồm hàng nghìn thành phần như: các transistor hoặc các cổng logic, cho nên với hệ thống số lớn, thiết kế ở mức cổng không còn sử dụng nữa. Qua nhiều thập kỷ, giản đồ logic của các thiết kế logic cũng không còn nhiều nữa. Ngày nay, sự phức tạp của phần cứng đã tăng lên ở một mức độ mà giản đồ của cổng logic hầu như vô ích khi nó chỉ biểu diễn một mạng lưới phức tạp các liên kết không theo chức năng của thiết kế. Từ những năm 1970, các kỹ sư điện và máy tính đổi hướng theo ngôn ngữ mô tả phần cứng (HDL). Hai ngôn ngữ mô tả phần cứng nổi bật trong kỹ thuật là Verilog và VHDL nhưng những nhà thiết kế công nghệ thích sử dụng Verilog hơn.

Cách sử dụng cơ bản của Verilog HDL trong thiết kế mạch tích hợp là mô phỏng thiết kế và tạo mẫu trên FPGA trước khi chuyển sang sản xuất. Mục tiêu của Verilog không phải tạo ra những chip VLSI mà sử dụng Verilog để mô tả một cách chính xác chức năng của bất kỳ hệ thống số nào và nạp chương trình tạo mẫu lên FPGA, ví dụ: như máy tính, các bộ vi xử lý,... tuy tốc độ chậm và lãng phí diện tích hơn. Những thiết kế mức thấp hơn trong Verilog được thực hiện trên VLSI để đạt đến tốc độ cực đại và có diện tích cực tiểu. Tuy nhiên sử dụng thiết kế dùng Verilog trên FPGA sẽ tiết kiệm chi phí và thời gian thiết kế.

2.1.2. Một số quy ước thiết kế

- Các quy ước gắn với ngôn ngữ lập trình C. Lời giải thích được đặt ở cuối câu bởi dấu “//” hay kí hiệu /* . . . */. Các từ khóa được dành riêng và tất cả ký tự là chữ thường. Ngôn ngữ phân biệt chữ in và chữ thường.

Một số có thể có dấu hoặc không dấu được xác định theo công thức

<size><base format><number>

<size> xác định số bit. <base format> là ký tự đơn “ ‘ ”, theo sau là một trong các ký tự b (nhị phân), d (thập phân), o (bát phân) và h (thập lục phân). <number> chứa các chữ số phù hợp.

Ví dụ :

‘h8fa // số hex ‘o765 // số bát phân

4'b1101 // số nhị phân 4 bit 1101

- Cấu trúc chương trình

Module: ngôn ngữ Verilog mô tả hệ thống số như là thiết lập một module Cấu trúc một module như sau :

module module_name (danh sách các port);

//các khai báo

input, output, inout, reg, wire, parameter . . .

//các câu lệnh Initial statement Always statement

Module Instantiation

Continuous assignment endmodule

Ví dụ : Thiết kế mô hình hành vi cho cổng NAND module NAND (in1, in2, out);

input in1, in2; output out;

assign out=~(in1&in2) endmodule

Với & là toán tử “and”, ~ là toán tử đảo bit. Điều lệnh assign chờ sự thay đổi ở vế phải biểu thức và kết quả được gán cho biểu thức bên trái (ngõ ra out).

2.2. Công nghệ FPGA

2.2.1. Công nghệ FPGA là gì

FPGA là viết tắt của "Field Programmable Gate Array", là vi mạch dùng cấu trúc mảng phần tử logic mà người dùng có thể lập trình được, có thể thực hiện các tính năng logic thông qua máy tính của mình với giá rẻ, và có thể xóa đi viết lại nhiều lần. Xilinx là công ty đầu tiên bán ra FPGA và luôn chiếm thị phần lớn nhất. Ngoài ra còn có Altera, Lattice, Actel, QuickLogic. Nhiều nhà sản xuất hiện đang cung cấp rất nhiều cấu trúc và phương pháp xử lý khác nhau. Vì vậy, việc chọn một cấu trúc và một phương pháp xử lý để có thể đáp ứng được các yêu cầu về chức năng và độ ổn định, đối với một hệ thống trong thực tế là vấn đề mang tính quyết định. Thiết bị logic lập trình được được phát minh lần đầu vào cuối thập kỷ 1970 và ngay lập tức đã trở lên phổ biến trong ngành công nghiệp bán dẫn. Ngoài khả năng lập trình đa dạng, công nghệ này còn có thời gian sản xuất nhỏ nên có khả năng cạnh tranh thương mại rất lớn. Hơn nữa, việc thiết kế với nó khá dễ dàng và nó có khả năng lập trình lại nhiều lần.

2.2.2. Tại sao công nghệ FPGA được chú ý

Sự xuất hiện của các thiết bị có thể lập trình như FPGA (Field Programmable Gate Array) đã tạo ra một bước đột phá trong việc xây dựng các hệ thống tích hợp (SoC -

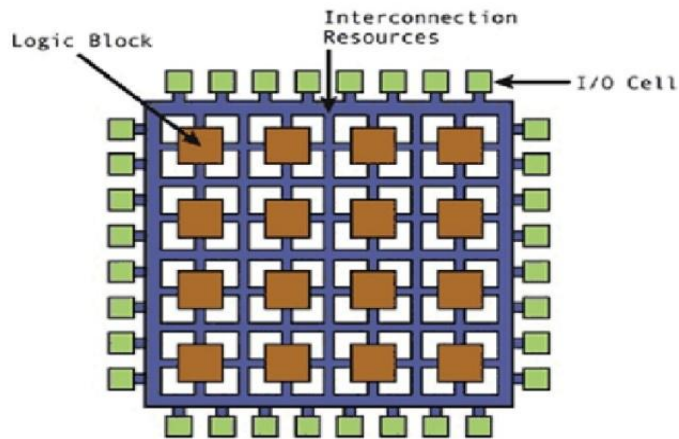
System on Chip). Với công nghệ FPGA, đã có rất nhiều công trình nghiên cứu để giải quyết các bài toán có yêu cầu khối lượng tính toán lớn như các bài toán về xử lý tín hiệu số, mật mã và nhận dạng. Do khả năng tái cấu trúc đơn giản và sở hữu một khối tài nguyên logic lớn FPGA có thể được ứng dụng cho nhiều các lớp bài toán

xử lý tín hiệu số cỡ lớn mà các công nghệ trước đó không làm được hoặc làm được nhưng với tốc độ và hiệu suất thấp.

2.2.3. Kiến trúc chung của một FPGA

Cấu trúc tổng thể của FPGA bao gồm:

- Các khối Logic.
- Hệ thống liên kết mạch.
- Các phần tử tích hợp sẵn.



Hình 2 - 1: Cấu trúc tổng thể của một FPGA

2.2.4. So sánh CPLD và FPGA

FPGA được thiết kế đầu tiên bởi Ross Freeman, người sáng lập công ty Xilinx vào năm 1984, kiến trúc mới của FPGA cho phép tích hợp số lượng tương đối lớn các phần tử bán dẫn vào 1 vi mạch so với kiến trúc trước đó là CPLD. FPGA có khả năng chứa tới từ 100.000 đến hàng vài tỷ cổng logic, trong khi CPLD chỉ chứa từ 10.000 đến 100.000 cổng logic, con số này đối với PAL, PLA còn thấp hơn nữa chỉ đạt vài nghìn đến 10.000.

CPLD được cấu trúc từ số lượng nhất định các khối SPLD (Simple programmable devices, thuật ngữ chung chỉ PAL, PLA). SPLD thường là một mảng logic AND/OR lập trình được có kích thước xác định và chứa một số lượng hạn chế các

phần tử nhớ đồng bộ (clocked register). Cấu trúc này hạn chế khả năng thực hiện những hàm phức tạp và thông thường hiệu suất làm việc của vi mạch phụ thuộc vào cấu trúc cụ thể của vi mạch hơn là vào yêu cầu bài toán.

Kiến trúc của FPGA là kiến trúc mảng các khối logic, khối logic, nhỏ hơn nhiều nếu đem so sánh với một khối SPLD, ưu điểm này giúp FPGA có thể chứa nhiều hơn các phần tử logic và phát huy tối đa khả năng lập trình của các phần tử logic và hệ thống mạch kết nối, để đạt được mục đích này thì kiến trúc của FPGA phức tạp hơn nhiều so với CPLD.

Một điểm khác biệt với CPLD là trong những FPGA hiện đại được tích hợp nhiều những bộ logic số học đã sơ bộ tối ưu hóa, hỗ trợ RAM, ROM, tốc độ cao hay các bộ nhân cộng (multiplication and accumulation, MAC), thuật ngữ tiếng Anh là DSP slice dùng cho những ứng dụng xử lý tín hiệu số DSP.

2.3. Kiến trúc của MIPS

2.3.1. Giới thiệu

MIPS (Microprocessor without Interlocked Pipeline Stage) là một kiến trúc xử lý được phát triển bởi hãng MIPS Technologies. Năm 1981, John L. Hennessy đứng đầu một nhóm bắt đầu một công trình nghiên cứu về bộ xử lý MIPS đầu tiên tại Stanford University. Nhóm nghiên cứu Stanford có một nền tảng mạnh mẽ trong biên dịch đã đưa họ đến sự phát triển một bộ xử lý, làm giảm bớt sự có mặt của trình biên dịch ở cấp phần cứng, trái ngược với sự tăng lên của phần cứng với phần mềm, nó là một bước tiến dài trong triết lý thiết kế phần cứng. Một thiết kế chủ chốt trong MIPS là yêu cầu các câu lệnh phải hoàn thành trong 1 chu kỳ máy. Cuối những năm 1990, 1/3 số lượng chip RISC được sản xuất dựa trên kiến trúc MIPS. Thiết kế MIPS rất giống với thiết kế RISC điển hình. Để tiết kiệm bit trong vi lệnh, thiết kế RISC giảm số lượng vi lệnh. Thiết kế MIPS sử dụng 6 bit của 32 bit cho vi lệnh cơ bản, còn lại là 26 bit là địa chỉ nhảy hoặc sử dụng 3 trường 5 bit để chỉ định

lên đến 3 thanh ghi với giá trị dịch kết hợp với 6 bit vi lệnh mở rộng hoặc một định dạng khác sử dụng 2 thanh ghi với một giá trị 16 bit mã trong vi lệnh.

Kiến trúc MIPS chỉ cho phép truy cập bộ nhớ thông qua các lệnh load và store, tất cả các lệnh còn lại đều sử dụng toán tử thanh ghi.

+ **Ưu điểm:** Bộ xử lý thanh ghi nhanh nhất (hơn 1 tỷ lần trong 1 giây) vì thanh ghi là một thành phần phần cứng thường nằm chung mạch với bộ xử lý. Có nhiều tập lệnh với hai trường vi lệnh.

+ **Khuyết điểm:** Do thanh ghi là một thành phần phần cứng nên số lượng cố định và hạn chế. Do đó, sử dụng phải khéo léo.

Các thế hệ của MIPS:

- + Ban đầu MIPS là kiến trúc 32 bit, sau này mở rộng ra 64bit.
- + MIPS I, MIPS II, MIPS III, MIPS IV, MIPS V, MIPS 32 và MIPS 64. Hiện nay tồn tại MIPS 32 và MIPS 64.
- + Các dòng vi xử lý thương mại MIPS đã được sản xuất:
 - Năm 1985, R2000.
 - Năm 1988, R3000.
 - Năm 1991, R4000 mở rộng tập lệnh đầy đủ cho 64bit, 100MHz, 8kB.
 - Năm 1993, R4400 16kB.
 - Năm 1994, R8000 thiết kế superscalar đầu tiên của MIPS.

2.3.2. Tập lệnh MIPS (một phần)

Trong nội dung thiết kế, chỉ một số lệnh phổ biến trong tập lệnh đầy đủ của MIPS 32 bit được hỗ trợ. Các lệnh này thuộc đầy đủ các định dạng nhóm lệnh cần có trong MIPS32. Các lệnh chưa được hỗ trợ khi cần có thể bổ sung nhanh chóng theo nhóm tương ứng. Vì thế việc nâng cấp thiết kế sau này được thực hiện dễ dàng.

