



TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI

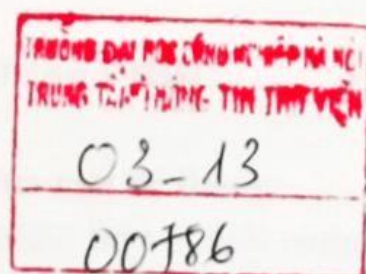
GIÁO TRÌNH NGÔN NGỮ MÔ TẢ PHẦN CỨNG



NHÀ XUẤT BẢN GIÁO DỤC VIỆT NAM

TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI
NGUYỄN ANH DŨNG – TỐNG VĂN LUYỀN – NGUYỄN VIỆT TUYẾN

GIÁO TRÌNH
NGÔN NGỮ MÔ TẢ PHẦN CỨNG



NHÀ XUẤT BẢN GIÁO DỤC VIỆT NAM

MỤC LỤC

Lời nói đầu	5
Chương 1. Tổng quan về ngôn ngữ mô tả phần cứng VHDL	7
1.1. Giới thiệu chung	7
1.2. Các phương pháp thiết kế tiền VHDL	7
1.2.1. Phương pháp thiết kế dùng hàm logic	7
1.2.2. Phương pháp thiết kế dựa trên sơ đồ nguyên lý	8
1.2.3. Nhược điểm của các phương pháp thiết kế truyền thống.....	9
1.3. Ngôn ngữ mô tả phần cứng	10
1.3.1. Lịch sử ngôn ngữ VHDL	11
1.3.2. Các chuẩn của ngôn ngữ VHDL.....	12
1.3.3. Lợi ích của ngôn ngữ VHDL.....	12
Câu hỏi và bài tập chương 1	13
Chương 2. Mô tả hệ thống bằng VHDL	14
2.1. Giới thiệu chung	14
2.2. Các thành phần cơ bản của một hệ thống	14
2.2.1. Hệ thống trao đổi thông tin với môi trường bên ngoài	14
2.2.2. Phần thân của hệ thống	15
2.2.3. Hỗ trợ từ bên ngoài	15
2.2.4. Thực thể xác định giao tiếp hệ thống.....	16
2.2.5. Kiến trúc mô tả phần thân	17
2.2.6. Gói hỗ trợ các đơn vị thiết kế khác.....	20
2.3. Tín hiệu	22
2.3.1. Giới thiệu về tín hiệu	22
2.3.2. Khai báo tín hiệu.....	23
2.4. Giao tiếp hệ thống.....	26
2.4.1. Tiêu đề của thực thể (Entity Header).....	26
2.4.2. Mệnh đề Port	29
2.4.3. Mệnh đề Generic.....	30
Câu hỏi và bài tập chương 2	31

Chương 3. Kiến trúc của chương trình VHDL	34
3.1. Giới thiệu chung	34
3.2. Các loại dữ liệu trong VHDL	34
3.2.1. Các kiểu dữ liệu tiền định nghĩa.	34
3.2.2. Các kiểu dữ liệu người dùng định nghĩa	37
3.2.3. Kiểu mảng	39
3.2.4. Kiểu bản ghi (Record)	41
3.2.5. Kiểu dữ liệu có dấu và không dấu (Signed and Unsigned)	42
3.2.6. Chuyển đổi dữ liệu	44
3.3. Toán tử, toán hạng và hằng số	46
3.3.1. Toán tử	46
3.3.2. Thuộc tính	53
3.3.3. Hằng số (CONSTANT)	55
3.3.4. Biến (VARIABLE)	55
3.4. Quá trình và hoạt động của quá trình	60
3.4.1. Lệnh WAIT	61
3.4.2. Lệnh IF	63
3.4.3. CASE	65
3.4.4. LOOP	69
3.5. Đa quá trình trong một kiến trúc	76
3.6. Các quá trình đơn giản	77
3.7. Một số đặc điểm khác	79
Câu hỏi và bài tập chương 3	80
Chương 4. Cấu trúc hệ thống và kiểm tra thiết kế	82
4.1. Mô tả cấu trúc	82
4.2. Cài đặt thực thể trực tiếp	84
4.3. Các thành phần và cấu hình	85
4.4. Kiểm tra thiết kế với TEST BENCH	88
Phụ lục	90
Tài liệu tham khảo	98

LỜI NÓI ĐẦU

Để phục vụ công tác giảng dạy môn học Ngôn ngữ mô tả phần cứng VHDL, Bộ môn Điện tử máy tính biên soạn cuốn giáo trình "Ngôn ngữ mô tả phần cứng VHDL". Giáo trình trình bày một cách hệ thống những kiến thức từ cơ bản đến nâng cao, nhằm mục đích giới thiệu cho sinh viên, những kiến thức cơ bản về ngôn ngữ mô tả phần cứng VHDL và từng bước làm làm chủ VHDL trong thiết kế các hệ thống số. Nội dung giáo trình gồm 4 chương:

Chương 1. Tổng quan về VHDL.

Chương 2. Mô tả hệ thống bằng VHDL.

Chương 3. Kiến trúc của chương trình VHDL.

Chương 4. Cấu trúc hệ thống và kiểm tra thiết kế.

Giáo trình này được sử dụng để giảng dạy cho sinh viên Cao đẳng và Đại học ngành Công nghệ kỹ thuật điện tử Trường Đại học Công nghiệp Hà Nội. Với kết cấu các chương mục một cách logic, rõ ràng, chúng tôi hy vọng giáo trình giúp được sinh viên tiếp thu nhanh chóng các kiến thức cơ bản ngôn ngữ mô tả phần cứng VHDL. Mặc dù đã hết sức cố gắng trong quá trình biên soạn, nhưng khó tránh khỏi thiếu sót, rất mong nhận được ý kiến đóng góp của các em sinh viên và các đồng nghiệp.

Thư góp ý xin gửi về:

*Bộ môn Điện tử máy tính, Khoa Điện tử - Trường Đại học Công nghiệp Hà Nội
(km 13, Minh Khai, Từ Liêm, Hà Nội).*

Các tác giả

Chương 1

TỔNG QUAN VỀ NGÔN NGỮ MÔ TẢ PHẦN CỨNG VHDL

1.1. GIỚI THIỆU CHUNG

Đối với hầu hết những người có ý định học ngôn ngữ mô tả phần cứng VHDL, câu hỏi đầu tiên xuất hiện trong đầu họ là: Tại sao chúng ta cần học ngôn ngữ mô tả phần cứng VHDL (Very high speed integrated circuit Hardware Description Language - ngôn ngữ mô tả phần cứng cho các vi mạch có tốc độ rất cao)? Chương này sẽ từng bước giúp ta trả lời câu hỏi đó thông qua các nội dung cơ bản:

Phần một tóm lược về hai phương pháp thiết kế truyền thống tiền VHDL: Phương pháp thiết kế logic dùng “bút chì và giấy” với các hàm logic (Boolean equations) và phương pháp thiết kế ưu việt hơn – phương pháp thiết kế dựa trên sơ đồ nguyên lý có sự trợ giúp của máy tính (computer – aided schematic – based design). Cả hai phương pháp thiết kế truyền thống này hiện vẫn đang được sử dụng nhưng tồn tại rất nhiều nhược điểm và không đáp ứng được yêu cầu thiết kế mạch ngày càng đa dạng và phức tạp hiện nay. Ngôn ngữ mô tả phần cứng (HDL - Hardware Descriptions Languages) ra đời đã khắc phục được nhược điểm của các phương pháp thiết kế truyền thống và đã chứng tỏ là phương pháp thiết kế mạch ưu việt trong thời đại mới.

Nội dung chương cũng giới thiệu sơ lược về các ngôn ngữ mô tả phần cứng phổ biến nhất được sử dụng trong việc thiết kế các mạch logic lập trình được (Programmable Logic Design) và giải thích tại sao chúng có thể đáp ứng được yêu cầu của công nghệ hiện nay.

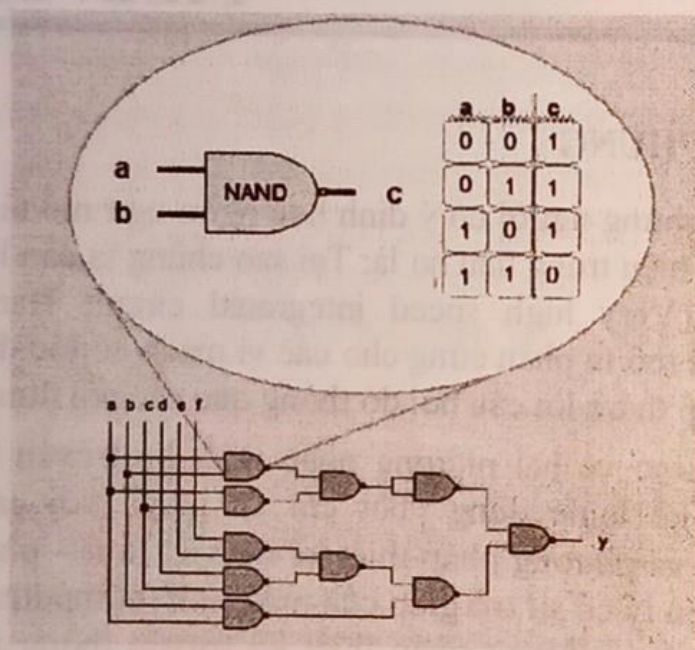
Mục sau giới thiệu về VHDL với vai trò là ngôn ngữ mô tả phần cứng hàng đầu của thập kỷ 90 thế kỷ XX bao gồm lịch sử phát triển VHDL, các khái niệm cơ bản của VHDL và các ưu điểm và nhược điểm của VHDL. Mục này giúp người học có cái nhìn tổng quan và khách quan về VHDL để dễ dàng tập trung tìm hiểu chi tiết và đầy đủ về VHDL được trình bày trong toàn bộ giáo trình này.

1.2. CÁC PHƯƠNG PHÁP THIẾT KẾ TIỀN VHDL

1.2.1. Phương pháp thiết kế dùng hàm logic

Trong việc thiết kế các hệ thống số, sẽ rất khó khăn nếu người thiết kế không có những kiến thức cơ bản về đặc điểm và chức năng của các phần tử logic cơ bản

như các cổng logic AND, OR, NOT,... cũng như các flip-flop. Hầu hết các mạch logic tạo nên nhờ các cổng logic và các flip-flop được thiết kế theo phương pháp truyền thống dựa trên các hàm logic (boolean equations). Nhiều kỹ thuật thiết kế đã ra đời nhằm tối ưu hóa phương pháp truyền thống này, một trong các công việc để tối ưu phương pháp thiết kế dùng hàm logic là làm giảm thiểu các phương trình logic giúp sử dụng các cổng logic và các flip-flop hiệu quả hơn.



Hình 1.1. Thiết kế mạch dùng hàm luận lý

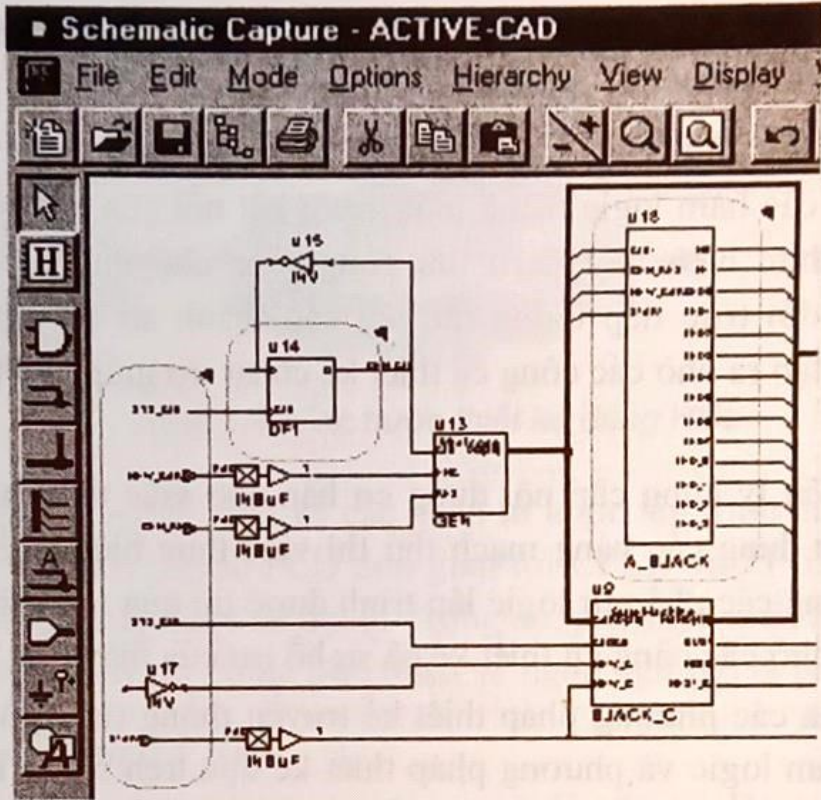
Kỹ thuật thiết kế dựa trên các hàm logic yêu cầu phải viết các phương trình logic cho từng đầu vào dữ liệu của flip-flop và cho từng nhóm cổng logic. Điều này có nghĩa kỹ thuật thiết kế sẽ không khả thi với việc thiết kế các mạch lớn với hàng trăm các flip-flop bởi vì kỹ thuật này thậm chí đòi hỏi phải có một số lượng lớn hơn hàng trăm các phương trình logic tương ứng.

Về lý thuyết, bất kỳ một hệ thống nào cũng có thể được biểu diễn bởi các hàm logic. Tuy nhiên, trên thực tế với các hệ thống số ngày nay với mức độ tích hợp và sự phức tạp đòi hỏi tới hàng nghìn các hàm logic thì phương pháp thiết kế dùng hàm logic là không khả thi.

1.2.2. Phương pháp thiết kế dựa trên sơ đồ nguyên lý

Phương pháp thiết kế dựa trên sơ đồ nguyên lý có sự trợ giúp của máy tính cho phép thiết kế các hệ thống lớn hơn nhờ kết hợp các cổng logic và các flip-flop với các mạch. Bởi vì các mạch có thể bao gồm rất nhiều các flip-flop cũng như các mạch khác, điều này cho phép thiết kế các mạch lớn có tính phân cấp với số

lượng lớn các thành phần tạo nên mà không cần mất nhiều công sức như phương pháp thiết kế dùng hàm logic trước đây.



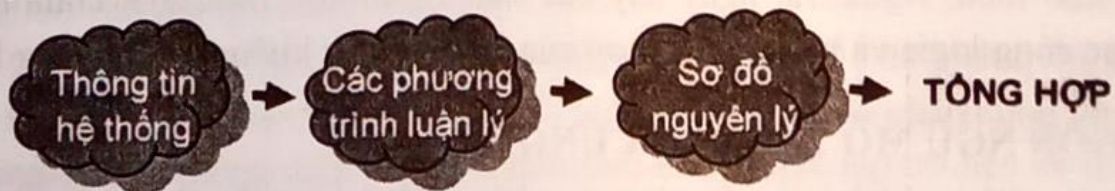
Hình 1.2. Thiết kế mạch dựa trên sơ đồ nguyên lý

Mọi người ưa thích và thường xuyên sử dụng phương pháp này còn bởi khả năng biểu diễn thiết kế trực quan sinh động các thiết kế bằng các sơ đồ nguyên lý dễ hiểu về các thành phần thiết kế và kết nối giữa chúng.

Trong nhiều năm, phương pháp thiết kế dựa trên sơ đồ nguyên lý có sự trợ giúp của máy tính được coi là sự lựa chọn tối ưu cho việc thiết kế. Ngày nay, với sự phát triển nhanh chóng của các thiết bị và hệ thống điện tử, phương pháp thiết kế này không còn đáp ứng tốt và tốn kém thời gian.

1.2.3 Nhược điểm của các phương pháp thiết kế truyền thống

Trước khi nhận xét về đặc điểm của các phương pháp thiết kế truyền thống, chúng ta điểm lại một số nội dung cơ bản của các phương pháp thiết kế này thông qua các bước cơ bản sau:



Hình 1.3. Các bước thiết kế của phương pháp thiết kế truyền thống

Bước 1:

Thông tin về hệ thống bao gồm mô tả đặc trưng và chức năng của hệ thống cần thiết kế được chuyển đổi thủ công sang một tập các hàm logic và sau đó là sơ đồ nguyên lý được tạo ra.

Bước 2:

Chuyển đổi các hàm logic thành một mạng kết nối của các cổng logic và các flip-flop được thực hiện một cách thủ công (các chuyên gia có kinh nghiệm thường chuyển đổi trực tiếp thông tin yêu cầu thành sơ đồ nguyên lý). Sơ đồ nguyên lý được tạo ra nhờ các công cụ thiết kế có sự trợ giúp của máy tính.

Bước 3:

Sơ đồ nguyên lý cung cấp nội dung cơ bản cho việc tổng hợp và thực hiện thiết kế. Nếu sử dụng các bảng mạch thử thì việc thực hiện thiết kế diễn ra thủ công, nếu sử dụng các phần tử logic lập trình được thì quá trình thực hiện thiết kế được thực hiện nhờ các công cụ thiết kế có sự hỗ trợ của máy tính.

Ưu điểm của các phương pháp thiết kế truyền thống (bao gồm phương pháp thiết kế dùng hàm logic và phương pháp thiết kế dựa trên sơ đồ nguyên lý) là dễ sử dụng. Tuy nhiên các phương pháp thiết kế bộc lộ nhiều nhược điểm. Một đặc điểm quan trọng nhất của các phương pháp này là một hệ thống luôn luôn được xác định như là một mạng kết nối của các thành phần chứ không phải là đặc điểm kỹ thuật của hệ thống được tạo ra và cung cấp cho ta. Đặc điểm kỹ thuật của một hệ thống luôn được thể hiện dưới dạng “hành vi/chức năng” cần có của hệ thống (ví dụ: hệ thống sẽ làm gì trong từng tình huống cụ thể).

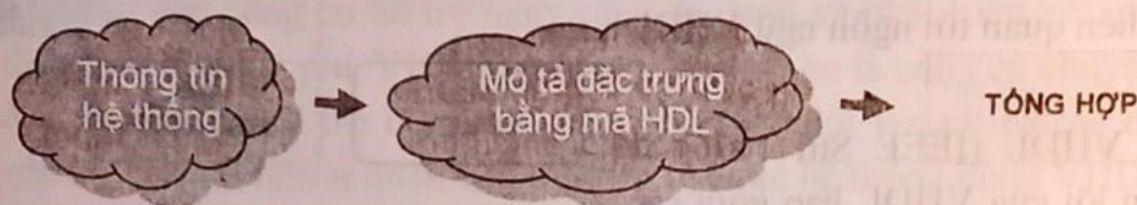
Một trong những nhược điểm của các phương pháp thiết kế truyền thống là khi sử dụng để thiết kế các mạch tích hợp lớn và phức tạp. Các mạch với hàng trăm các phương trình đã là bài toán khó nhưng vẫn còn khả thi. Tuy nhiên thật khó tưởng tượng ta sử dụng các phương pháp thiết kế truyền thống như thế nào khi thiết kế các mạch với hàng nghìn các phương trình.

Đối với phương pháp thiết kế dựa trên mạch nguyên lý đã được đề cập ở trên, khi thiết kế các mạch với hơn 6000 cổng logic kết quả thu được trở nên quá phức tạp và khó hiểu. Ngoài ra, ngày nay các mạch tích hợp mới nhất chứa tới hàng triệu các cổng logic và mật độ tích hợp của chúng đang không ngừng tăng lên.

1.3. NGÔN NGỮ MÔ TẢ PHẦN CỨNG

Nhược điểm chính của các phương pháp thiết kế truyền thống là công đoạn chuyển đổi thủ công từ mô tả thông tin thiết kế thành các phương trình logic.

Bước này có thể hoàn toàn được loại bỏ nếu dùng HDL. Ví dụ, hầu hết các công cụ HDL cho phép sử dụng mô hình máy trạng thái hữu hạn FSM (Máy trạng thái hữu hạn - Finite State Machines) cho các bảng chân lý và hệ thống tuần tự để kết hợp các module. Những mô tả thông tin thiết kế này có thể được chuyển đổi tự động thành các mã HDL mà từ đó ta có thể thực hiện được thiết kế bằng các công cụ tổng hợp.



Hình 1.4. Các bước thiết kế dùng HDL

HDL được sử dụng để thiết kế các phần tử logic lập trình được (Programmable Logic Devices – PLD) từ các PLD đơn giản tới CPLD và FPGA phức tạp. Hiện nay có nhiều HDL đang được sử dụng. Trong đó, phổ biến nhất là VHDL, Verilog và Abel. Giáo trình này tập trung giới thiệu về ngôn ngữ mô tả phần cứng VHDL.

1.3.1. Lịch sử ngôn ngữ VHDL

VHDL (Very high speed integrated circuit Hardware Description Language) là một trong các ngôn ngữ mô tả phần cứng được sử dụng rộng rãi hiện nay. VHDL là ngôn ngữ mô tả phần cứng cho các vi mạch tích hợp có tốc độ cao, được phát triển dùng cho chương trình VHSIC (Very High Speed Integrated Circuit) của Bộ Quốc phòng Mỹ.

Mục đích của việc nghiên cứu và phát triển là tạo ra một ngôn ngữ mô phỏng phần cứng chuẩn và thống nhất, cho phép thử nghiệm các hệ thống số nhanh hơn, hiệu quả hơn, và nhanh chóng đưa các hệ thống đó vào ứng dụng.

Tháng 7 năm 1983, ba công ty Internetic, IBM, Texas Instruments bắt đầu nghiên cứu. Sau một thời gian, phiên bản đầu tiên của ngôn ngữ VHDL được công bố vào tháng 8 năm 1985.

Vào năm 1986, VHDL được công nhận như một chuẩn IEEE. VHDL đã qua nhiều lần kiểm nghiệm và chỉnh sửa cho đến khi được công nhận như một chuẩn IEEE 1076 vào tháng 12 năm 1987.

VHDL được nghiên cứu phát triển nhằm giải quyết tốc độ phát triển, các thay đổi và xây dựng các hệ thống điện tử số. Với một ngôn ngữ phần cứng tốt thì việc xây dựng các hệ thống điện tử số có tính linh hoạt, phức tạp trở nên dễ dàng hơn. Việc mô tả hệ thống số bằng ngôn ngữ cho phép xem xét, kiểm tra toàn bộ hoạt động của hệ thống trong một mô hình thống nhất.

1.3.2. Các chuẩn của ngôn ngữ VHDL

Từ khi ra đời đến nay, ngôn ngữ VHDL không ngừng được nghiên cứu, chỉnh sửa và phát triển. Kết quả là chúng ta đã xây dựng được nhiều chuẩn có liên quan tới ngôn ngữ VHDL bao gồm:

1076 VHDL (IEEE Std 1076) là tiêu chuẩn lõi của VHDL bao gồm các định nghĩa về ngôn ngữ, cụ thể:

IEEE Std 1076.1 sử dụng cho thiết kế các hệ thống tương tự hoặc lai giữa tương tự và số.

IEEE Std 1076.2 hỗ trợ các tiện ích toán học cho các hoạt động toán học.

IEEE Std 1076.3 sử dụng cho việc tổng hợp các tập con VHDL.

IEEE Std 1076.4 hỗ trợ quá trình tiêu chuẩn hóa các mạch tích hợp ứng dụng chuyên biệt ASIC.

IEEE Std 1076.5 dành cho việc mở rộng và thống nhất các thành phần thư viện IEEE trong VHDL, thư viện 1 sử dụng cho thiết kế các hệ thống tương tự hoặc lai giữa tương tự và số.

1164 STD LOGIC (IEEE Std logic 1164) cho phép mở rộng VHDL với các logic đa giá trị cần thiết để mô tả hệ thống thời gian thực. Nó là tiêu chuẩn công nghiệp và được sử dụng bởi tất cả các công cụ VHDL.

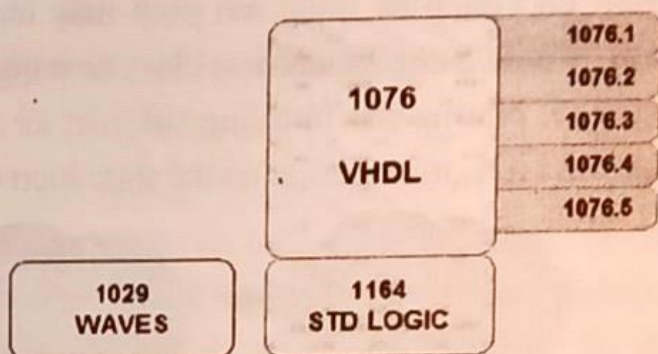
1029 WAVES (IEEE Std 1029 WAVES) (Đặc tả chuyển đổi dạng sóng và vector VHDL) định nghĩa định dạng dữ liệu để kiểm tra và kiểm nghiệm theo thời gian.

1.3.3. Lợi ích của ngôn ngữ VHDL

Hầu hết các nhà thiết kế hệ thống điện tử mong muốn sử dụng VHDL bởi tính ưu việt của ngôn ngữ này như:

VHDL là ngôn ngữ mô tả phần cứng thông dụng nhất trên toàn cầu và được hầu hết các nhà thiết kế mạch lựa chọn làm công cụ thiết kế hệ thống.

VHDL có khả năng mô tả đa dạng các loại hệ thống khác nhau. Trên thực tế VHDL có thể được sử dụng được ở cả hai dạng hoặc như là một ngôn ngữ có cấu trúc (cung cấp mã chương trình dạng văn bản của một sơ đồ nguyên lý) hoặc dạng ngôn ngữ theo hành vi/chức năng. Thậm chí, hai dạng thiết kế này còn có thể sử



Hình 1.5. Các chuẩn của VHDL

dụng kết hợp cùng nhau. Mặc dù VHDL có thể được sử dụng cho việc thiết kế các hệ thống với mức độ phức tạp khác nhau từ một transistor tới một hệ thống hoàn chỉnh và mọi việc vẫn giữ nguyên trong cùng một môi trường mô phỏng VHDL.

VHDL được tích hợp sẵn các công cụ mô phỏng hệ thống. Do tính phổ cập của hệ thống, các công cụ hỗ trợ ngày càng đa dạng và phát triển với chi phí thấp hoặc thậm chí là miễn phí. VHDL ngày nay không còn là công cụ chuyên biệt cho các công ty lớn, nó đã trở thành công cụ phổ biến đối với tất cả các công ty và thậm chí là với mọi người quan tâm tới việc thiết kế hệ thống dùng VHDL.

Nhiều hãng cung cấp phần mềm hỗ trợ quá trình tổng hợp mạch từ ngôn ngữ VHDL. Điều này góp phần làm tăng giá trị của VHDL và giảm thiểu chi phí thiết kế hệ thống.

VHDL không chỉ giới hạn ứng dụng trong lĩnh vực điện tử, nó đã trở thành ngôn ngữ mô hình và mô phỏng thống nhất cho các lĩnh vực khác như cơ điện tử, thủy lực, hóa học và các lĩnh vực khác.

CÂU HỎI VÀ BÀI TẬP CHƯƠNG 1

Bài 1: Nêu đặc điểm của các phương pháp thiết kế dùng hàm logic, dùng sơ đồ nguyên lý. Lấy ví dụ minh họa về các phương pháp thiết kế này dùng hàm logic và sơ đồ nguyên lý.

Bài 2: Trình bày các ưu điểm cơ bản của phương pháp thiết kế mới dùng VHDL so với các phương pháp thiết kế truyền thống.

Bài 3: Nêu đặc điểm của ngôn ngữ mô tả phần cứng VHDL.

Bản quyền thuộc Trường Đại học Công nghiệp Hà Nội.

*Công ty cổ phần Sách Giáo dục tại TP. Hà Nội – Nhà xuất bản Giáo dục Việt Nam
giữ quyền công bố tác phẩm*

GIÁO TRÌNH NGÔN NGỮ MÔ TẢ PHẦN CỨNG

Mã số: 7B011S1 - TTS

Số ĐKKH xuất bản: 1048-2011/CXB/7-1495/GD.

In 1.000 cuốn (QĐ85TK), khổ 17 x 24 cm, tại Công ty CP In Phú Thọ.

Số in: 733. In xong và nộp lưu chiểu tháng 10 năm 2011.