BỘ GIÁO DỤC & ĐÀO TẠO
TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHỆ
THÀNH PHỐ HỒ CHÍ MINH

Ths. NGUYỄN TRỌNG HẢI

TÓM TẮT BÀI GIẢNG

VERILOG

LƯU HÀNH NỘI BỘ
07/2005

CHƯƠNG 1
TỔNG QUAN

Verilog HDL là một trong hai ngôn ngữ mô phỏng phần cứng thông dụng nhất, được dùng trong thiết kế IC, ngôn ngữ kia là VHDL.

HDL cho phép mô phỏng các thiết kế dễ dàng, sửa chữa lỗi, hoặc thực nghiệm bằng những cấu trúc khác nhau. Các thiết kế được mô tả trong HDL là những kỹ thuật độc lập, dễ thiết kế, dễ tháo gỡ, và thường dể đọc hơn ở dạng biểu đồ, đặc biệt là ở các mạch điện lớn.

Verilog thường được dùng để mô tả thiết kế ở bốn dạng:

Thuật toán (một số lệnh giống ngôn ngữ C như: if, case, for,while...).

Chuyển đổi thanh ghi (kết nối bằng các biểu thức Boolean).

Các cổng kết nối(cổng: OR, AND, NOT...).

Chuyển mạch (BJT, MOSFET).

Ngôn ngữ này cũng chỉ rõ cách thức kết nối, điều khiển vào/ra trong mô phỏng.

Cấu trúc chương trình dùng ngôn ngữ Verilog

```
// Khai báo module
Module tên chương trình (tên biến I/O); // tên chương trình trùng tên file.v.
Input [msb:lsb] biến;
Output [msb:lsb] biến;

Reg [msb:lsb] biến reg;
Wire [msb: lsb] biến wire;

// Khai báo khối always, hoặc khối initial.
... các lệnh ...

Endmodule
```

Chương II
CHỨC NĂNG CÁC TỪ VỰNG TRONG VERILOG

Những tập tin văn bản nguồn Verilog bao gồm những biểu hiện thuộc tính từ vựng sau đây:

I. Khoảng trắng

Khoảng trắng ngăn những từ và có thể chứa khoảng cách, khoảng dài, dòng mớivà dạng đường dẫn. Do đó, một lệnh có thể đưa ra nhiều dòng phức tạp hơn mà không có những đặc tính đặc biệt.

II. Chú giải

Những chú giải có thể chỉ định bằng hai cách: (giống trong C/C++)

Chú giải được viết sau hai dấu gạch xiên (//). Được viết trên cùng một dòng.

Được viết giữa /* */, khi viết nhiều dòng chú giải.

III. Chữ số:

Lưu trữ số được định nghĩa như là một con số của các bit, giá trị có thể là: số nhị phân, bát phân, thập phân, hoặc thập lục phân.

Ví dụ: 3'b001, 5'd30 = 5'b11110,

16'h5ED4 = 16'd24276 = 16'b0101111011010100

IV. Từ định danh:

Từ định danh do người dùng quy định cho biến số, tên hàm, tên môđun, tên khối và tên trường hợp. Từ định danh bắt đầu bằng một mẫu tự hoặc đường gạch dưới '_' (không bắt đầu bằng một con số hoặc $) và kể cả mọi chữ số của mẩu tự, những con số và đường gạch dưới, từ định danh trong Verilog thì phân biệt dạng chữ.

V. Cú pháp:

Kí hiệu cho phép:

ABDCE...abcdef...1234567890_$

Không cho phép: các kí hiệu khác -, &, #, @

VI. Toán tử:

Toán tử là một, hai, hoặc ba kí tự dùng để thực hiện các toán hạng trên biến. Các toán tử bao gồm >, +, &, !=.

VII. Từ khóaVerilog:

Có những từ mà phải có ý nghĩa đặc biệt trong Verilog. Ví dụ: **assign, case, while, wire, reg, and, or, nand,** và **module**. Chúng không được dùng như từ định danh. Từ khóa Verilog cũng bao gồm cả chỉ dẫn chương trình biên dịch và System Task (hệ thống soạn thảo) và các hàm.

Chương III
CÁC CỔNG CƠ BẢN TRONG VERILOG

Các cổng logic cơ sở là một bộ phận của ngôn ngữ Verilog. Có hai đặc tính được chỉ rõ là: drive_strenght và delay.

Drive_strenght chỉ sức bền của cổng. Độ bền ngõ ra là sự kết nối một chiều đến nguồn, kế đó tạo nên sự kết nối trong suốt trans dẫn, kết thúc là tổng trở kéo lên hoặc xuống. Drive_strenght thường không được chỉ rõ, trong trường hợp này độ bền mặc định là strong1 và strong0 .

Delay: nếu delay không được chỉ rõ, thì khi đó cổng không có trì hoãn truyền tải; nếu có hai delay được chỉ định, thì trước tiên là miêu tả trì hoãn lên, thứ hai là trì hoãn xuống. Nếu chỉ có một delay được chỉ định, thì khi đó trì hoãn lên xuống là như nhau. Delay được bỏ qua trong tổng hợp. Phương pháp của sự trì hoãn chỉ định này là một trường hợp đặc biệt của "Parameterized Modules". Các tham số cho các cổng cơ sở phải được định nghĩa trước như delay.

I. Các cổng cơ bản:

Các cổng cơ bản có một ngõ ra, và có một hoặc nhiều ngõ vào. Trong các cổng, cú pháp cụ thể biểu diễn bên dưới, các từ khoá của các cổng: and, or, nand, nor.

1. Cú pháp:

GATE (drive_strength)#(delays)

Tên từ khóa cổng _tên (output, input_1, input_2, ..., input_N);

Delay: #(lên, xuống) hoặc #lên_và_xuống hoặc #(lên_và_xuống)

2. Ví dụ:

And c1 (o, a, b, c. d); // có 4 ngõ vào cổng And gọi là c1
c2 (p, f, g); // và 2 ngõ vào cổng and gọi là c2

Or #(4,3) ig (o, b, c); // cổng Or được gọi là ig, rise time = 4, fall time = 3

Xor #(5) xor1 (a, b, c); // sau 5 đơn vị thời gian thì a = b xor c

II. Cổng buf, not:

Các cổng này thực thi đệm và đảo theo theo thứ tự địmh sẳn. Chúng có một ngõ vào, hai hay nhiều ngõ ra. Cú pháp cụ thể biểu diễn ở bên dưới, từ khoá buf, not.

1. Cú pháp:

Tên từ khóa cổng _tên (output_1, output_2, ..., output_N, input);

2. Ví dụ:

Not #(5) not_1(a,c); // sau 5 đơn vị thời gian thì a = đảo c

Buf c1 (o, p, q, r, in); // bộ đệm 5 ngõ ra và 2 ngõ ra

c2 (p, f, g);