

BỘ SOẠN THẢO SƠ ĐỒ LOGIC SCHED TRÊN MÁY VI TÍNH IBM-PC

DẶNG VĂN ĐỨC

I - ĐẶT VẤN ĐỀ

Ngày nay trong các mạch logic người ta không chỉ thấy các vi mạch (IC) chuẩn mà còn thấy xuất hiện nhiều các vi mạch chuyên dụng (ASIC Application Specific Integrated Circuit). Chức năng của các vi mạch này là bao gồm nhiều chức năng của các IC chuẩn. Một trong cách chế tạo vi mạch chuyên dụng là đi từ các tế bào chuẩn. Mỗi công ty chế tạo vi điện tử (INTEL, NATIONAL SEMICONDUCTOR...) đều đưa ra thư viện tế bào riêng (xem [3] và [4]). Việc chế tạo vi mạch chuyên dụng bao gồm các bước chính sau:

- Xác định chức năng của các vi mạch.
- Thiết kế sơ đồ logic.
- Mô phỏng: kiểm tra chức năng logic, thời gian, xác định véc tơ mô phỏng lỗi...
- Xếp sắp tế bào chuẩn và đi dây.
- Chế tạo.

Trong các công đoạn trên đây, người sử dụng có thể thực hiện từ điểm a đến điểm d ngay trên máy tính của họ trước khi gửi đi chế tạo nhờ một công cụ phần mềm đặc biệt gọi là hệ tự động thiết kế vi mạch.

Trong khuôn khổ hợp tác với công ty Incosys (Tây Đức), chúng tôi đã đưa ra một vài công cụ giúp thực hiện một số bước trên máy vi tính IBM-PC.

II - BỘ SOẠN THẢO SƠ ĐỒ LÔGIC

Quá trình mô phỏng lôgic yêu cầu nhận biết danh sách nối các linh kiện (NETLIST) trong sơ đồ. Có nhiều phương pháp để nhận biết NETLIST của mạng. Song chúng được chia làm hai nhóm chính sau:

- Dịch thủ công sơ đồ logic trên giấy sang NETLIST. Nhờ chương trình soạn thảo văn bản để nhập chúng vào máy trước khi mô phỏng. Phương pháp này dễ gây ra nhầm lẫn.
- Dùng bộ chương trình có khả năng vẽ sơ đồ logic rồi tự động dịch sang NETLIST

Bộ soạn thảo sơ đồ lôgic SCHED trình bày ở đây thuộc họ này.

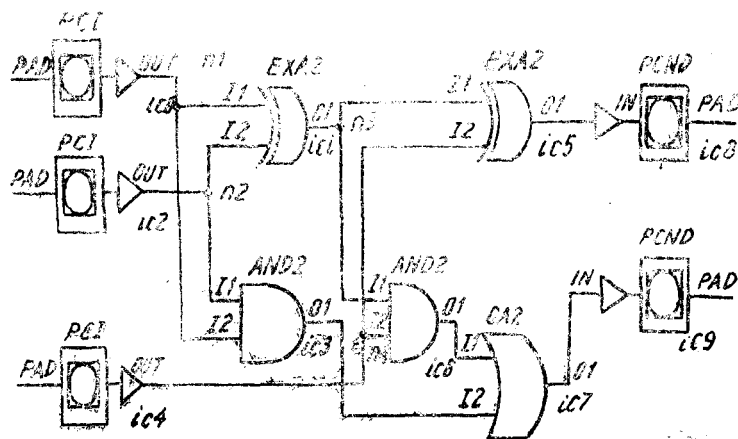
Trước khi mô tả hệ thống SCHED, chúng tôi trình bày một thí dụ để minh họa khái niệm NETLIST và nhiệm vụ của SCHED.

Giả sử chúng ta phải kiểm nghiệm hoạt động logic của bộ cộng (ADDER) mà chúng có sơ đồ lôgic sau:

SCHED Version 2/89. IIC - Hà Nội.

File's name: adder. sch.

Trong đó, các phần tử trong mạng là các phần tử chuẩn trong thư viện của hãng INTEL (xem [3]). Để kiểm nghiệm được sơ đồ trên, một trong các phương pháp là thực hiện mô phỏng chúng trên máy tính bằng hệ chương trình mô phỏng. Hệ chương trình này nhận biết sơ đồ trên qua quan hệ (nối) giữa các thành phần trong mạng. Chúng được biểu hiện bằng qui tắc sau:



node: { io [,] type [,] name ; } { pad }

trong đó:

node: tên của các điểm nối (nút) trong sơ đồ, thí dụ n1, n2... trong sơ đồ trên. Nếu một nút nào đó trong sơ đồ mà không có tên thì chúng tự động nhận tên mới.

io: tên đầu vào hoặc đầu ra của linh kiện trong mạch, thí dụ OUT, I1, O1...

name: tên linh kiện, thí dụ ico, ic1...

type: kiểu linh kiện, thí dụ PCI, ADN2, EXR2...

{ } : số lần lặp lại tùy ý.

[] : ký tự trong ngoặc vuông có thể thay thế kí tự đứng trước.

pad: tên đầu vào hoặc đầu ra của vi mạch, (Trong chế tạo vi mạch bán thành phẩm, đầu vào và đầu ra của vi mạch được coi như linh kiện có thể)

Từ sơ đồ của bộ cộng trên đây theo qui tắc này ta có:

S_{n1}: I1, EXR2, ic1; OUT PCI ico; I2 ADN2 ic3

(dòng trên đây cũng có thể viết như sau:

S_{n1}: I1, EXR2, ic1; OUT, PCI, ico; I2, ADN2, ic3)

S_{n2}: I2, EXR2 ic1; OUT PCI, ic2; I1 ADN2 ic3

S_{n3}: O1 EXR2 ic1; I1 EXR2 ic5; I1 ADN2 ic6

S₀: PAD PCI ico

S₄: PAD PCI ic2

S_{n4}: I2 EXR2 ic5; OUT PCI ic4; I2 ADN2 ic6

S₂: O1 EXR2 ic5; IN PCNO ic8

S₃: PAD PCI ic4

S₄: I1 OR2 ic7; O1 ADN2 ic6

S₅: I2 OR2 ic7; O1 ADN2 ic3

S₆: O1 OR2 ic7; IN PCNO ic9

S₇: PAD PCNO ic8

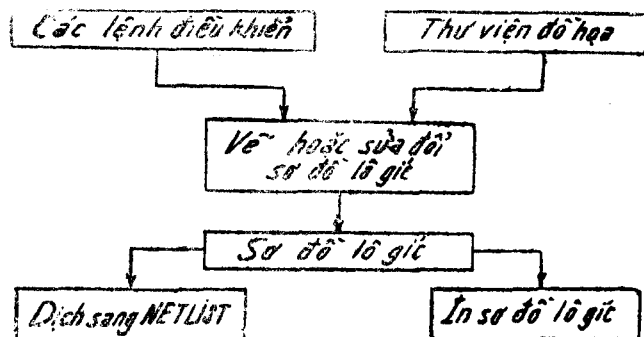
S₈: PAD PCNO ic9

Danh sách trên đây được gọi là NETLIST của bộ cộng. Mục tiêu của bộ soạn thảo sơ đồ logic SCHED là cho phép tự động lập danh sách như trên bằng máy tính.

1. Cấu trúc của hệ SCHED.

Cấu trúc tổng quát của hệ SCHED được mô tả trên hình dưới đây:

Nhờ bàn phím hoặc chuột, người sử dụng chọn các lệnh thích hợp để soạn thảo sơ đồ, quá trình vẽ từng linh kiện được gọi thực hiện nhanh chóng nhờ thư viện mô tả hình học các linh kiện. Kết quả của soạn thảo là sơ đồ có thể dịch sang NETLIST và in ra máy in đồ họa được.



2. Các chức năng của SCHED

2.1. Thư viện mô tả hình học các linh kiện

Thư viện của SCHED là tập hợp

các mô tả hình học để được linh kiện mong muốn. Việc mô tả được thực hiện bằng một trong hai phương pháp sau đây:

a) Lập ảnh linh kiện

Mỗi linh kiện được mô tả trong một ma trận điểm. Mỗi điểm của ma trận tương ứng với một điểm trên màn hình đồ họa. Các điểm được mã hóa 1 hoặc 0 để biểu thị hoặc không trên màn hình. Sau đây là thí dụ lập ma trận ảnh cho linh kiện khuếch đại (BUFFER). Linh kiện được vẽ trong ma trận 11 × 8.

01100000
 01010000
 01001000
 01000100
 01000010
 01000001
 01000010
 01000100
 01001000
 01010000
 01100000

Phương pháp này cho phép hiển thị khá nhanh các ảnh linh kiện trên màn hình, song nhược điểm của chúng là chiếm khá nhiều bộ nhớ. Chúng không được sử dụng trong hệ thống này.

b) Ngôn ngữ mô tả linh kiện

Chúng tôi đã xây dựng một ngôn ngữ kiểu ngôn ngữ CIF (Caltech Intermediat Form (xem [1]) để mô tả linh kiện. Một linh kiện, về mặt hình học, là tập hợp của các đoạn thẳng hình tròn hoặc cung tròn và các dòng văn bản làm tên, kiểu và các đầu vào ra của chúng. Mỗi thành phần của linh kiện được xác định nhờ các tọa độ tương đối. Chúng tôi đã định nghĩa cú pháp cho chúng để tránh độ phức tạp của CIF trong ứng dụng này. Đây là một ngôn ngữ đơn giản mỗi lệnh được biểu diễn trên một dòng. Sau đây là cú pháp của chúng:

Các ký hiệu sử dụng chung trong các dòng lệnh là:

{ } : số lần lặp lại tùy ý.

[] : ký tự trong dấu ngoặc nhọn có thể thay thế ký tự đứng trước đó.

text : dãy ký tự tùy ý.

Tọa độ được tính bằng số điểm trên màn hình, kể từ góc trên bên trái của ma trận điểm.

* text : sau dấu sao (*) là chú thích.

I{name [.,]} : danh sách các đầu vào.

O{name [.,]} : danh sách các đầu ra.

trong đó:

name là tên đầu vào hoặc đầu ra của linh kiện. P xo[.] yo[.] x1[.]y1

(xo, yo) tọa độ cực trên, bên trái của linh kiện.

(x1, y1) tọa độ cực dưới, bên phải của linh kiện.

N{xiO [.] yiO [.] xname [.] yname [.] text}

(xiO, yiO) tọa độ đầu vào/ra thứ i

(xname, yname) tọa độ của tên đầu vào/ra

text tên đầu vào/ra

L{xi, yi}

xi, yi tọa độ các đoạn thẳng hay đường gấp khúc

R{x, y, góc 1, góc 2, bán kính}

(x, y) tâm của đường tròn, hay cung tròn vẽ từ góc 1 đến góc 2

bán kính: bán kính của đường tròn hay cung tròn

T x, y, txt

(x, y) tọa độ của tên linh kiện

text tên linh kiện,

Thí dụ:

Mô tả linh kiện INVERTER bằng ngôn ngữ trên.

* Linh kiện đảo INVERTER

*-----

I (I1)

O (O1)

P 0, 0 50. 50

N 0, 20 0, 12 11 50, 20 35, 12 01

L 0, 2 17, 20

L 34, 20, 50, 20

L 17, 9 17, 31 28, 20 17, 9

R 32, 20, 300, 0, 2

T 5, 0 INVN

Đề hệ SCHED vẽ được linh kiện theo mô tả trên đây, chúng tôi đã xây dựng một trình dịch một bước với tên là CPL, chương trình CPL làm nhiệm vụ chuyển các dòng lệnh của ngôn ngữ mô tả thành cấu trúc dữ liệu riêng mà SCHED hiểu được. Tập hợp của các cấu trúc đó là thư viện đồ họa của SCHED.

Thư viện đồ họa của SCHED chứa hầu hết các tế bào chuẩn của INTER đưa ra trong [3]. Với cấu trúc này người sử dụng dễ dàng thêm bớt, thay đổi thư viện đồ họa của mình cho phù hợp với công nghệ chế tạo của từng hãng.

2.2. Điều khiển soạn thảo sơ đồ logic

Sched làm việc trên nguyên lý hỏi đáp qua các cửa sổ, người sử dụng đối thoại nhờ bàn phím hay chuột. Các chức năng chính gồm có:

- Thêm bớt, thay thế, chuyển dịch hay sao chép một hay một nhóm các linh kiện.
- Cho phép lập các điểm nối. Trong một sơ đồ logic thường phải nối các điểm ở cách xa nhau. Với khả năng này, người sử dụng chỉ cần lập các điểm nối mà không cần đi dây để nối chúng lại với nhau.

- Tự động lập và vẽ các NACRO. Số lượng linh kiện của một sơ đồ thường là rất lớn, để có thể vẽ chúng trên màn hình và mô phỏng, người sử dụng phải chia chúng thành từng phần với chức năng độc lập. Mỗi phần này được gọi là MACRO. Vậy, sơ đồ logic cuối cùng có thể là tập hợp các MACRO và các linh kiện chuẩn.

Một sơ đồ logic thường rất lớn so với màn hình vật lý của máy vi tính IBM/PC, trong thực tế ở hệ SCHED người sử dụng làm việc với màn hình đó với độ lớn gấp 16 lần. Thông tin mô tả toàn bộ sơ đồ được lưu trên đĩa dưới dạng tệp khá tối ưu nhờ tránh được kỹ thuật Bimap.

2.3. Dịch sơ đồ sang NETLIST

Trước khi chuyển sang giai đoạn mô phỏng trong quá trình chế tạo vi mạch bán thành phẩm, chúng ta phải chuyển tệp mô tả vừa vẽ sang NETLIST. Chúng cũng được ghi lên đĩa từ dưới dạng tệp.

Với phương pháp mô tả linh kiện và vẽ sơ đồ trình bày trên đây, chúng tôi đã đưa ra thuật toán thích hợp cho việc tự động lập danh sách nối giữa chúng như sau:

- a) Đặt mức 0 cho tất cả các đầu vào, đầu ra của các linh kiện.
- b) Xét duyệt lần lượt từng đầu vào / ra của mỗi linh kiện có mức 0, giả sử gọi i là mức hiện hành của chúng.
 - b1. Gán $1 \rightarrow i$ cho đầu vào hoặc đầu ra nếu thỏa mãn điểm b
 - b2. Gán $i+1 \rightarrow i$ cho các đầu vào / ra hoặc nút hoặc đoạn thẳng nối với đầu vào / ra có mức i
 - b3. Đánh dấu cùng mức cho các nút có cùng tên trong sơ đồ
 - b4. Tiếp tục lan truyền việc đánh dấu theo nguyên tắc trên đây cho các thành phần (đầu vào / ra, nút, đoạn thẳng) nối với các thành phần có mức đánh dấu nhỏ hơn một đơn vị.
 - b5. Việc đánh dấu kết thúc khi các nhánh lan truyền gặp đầu vào / ra của linh kiện.
- c) Các đầu vào / ra của linh kiện đã được đánh dấu (có mức khác 0 và -1) tập hợp thành một nút. Gán mức -1 cho các đầu vào / ra của nút vừa nhận được. Lập lại điểm b. Tập hợp các nút là danh sách nối các linh kiện (NETLIST) của sơ đồ đã cho.

2.4. In sơ đồ logic

Hệ SCHED sử dụng máy in đồ họa để in sơ đồ logic. Chúng ta có khả năng in từng trang sơ đồ mong muốn.

Hệ soạn thảo sơ đồ logic SCHED được viết bằng ngôn ngữ C. Có thể thực hiện chúng trên các máy vi tính IBM P/CXT/AT hoặc tương thích có màn hình đồ họa (CGA, EGA, VGA, HERCULES...) bộ nhớ RAM tối thiểu 640 kbytes. Có thể sử dụng chuột, MS MOUSE hoặc tương thích.
(Xem tiếp bìa 3)

BỘ SOẠN THẢO SƠ ĐỒ...

(Tiếp theo trang 3?)

III - KẾT LUẬN

Trên thị trường không có nhiều hệ soạn thảo sơ đồ logic cho chế tạo vi mạch chuyên dụng trên máy vi tính IBM-PC mà phần lớn các công cụ CAE/CAD là hướng tới sản xuất mạch in (EE Designer, OrCAD, E/Z CAD, ...). Do các hạn chế của máy vi tính IBM-PC về tốc độ, dung lượng bộ nhớ trong, màn hình đồ họa, phương pháp thực hiện các công cụ giúp mô phỏng các mạch logic trình bày trên đây là thích hợp.

Đặc tính chủ yếu của hệ SCHED vừa trình bày trên đây là dễ sử dụng, đủ để cho phép thực hiện bước tiếp theo của quá trình chế tạo vi mạch bán thành phẩm là mô phỏng logic. Vì toàn bộ chương trình được viết bằng ngôn ngữ C nên dễ thích nghi trên các hệ máy tính khác nhau. Bộ chương trình SCHED được thực hiện với sự cộng tác của K.S. Phan Minh Tân và các đồng nghiệp khác trong nhóm nghiên cứu về ASIC của viện KHTT và ĐK.

Nhận ngày 15-5-1989

TÀI LIỆU THAM KHẢO

1. Ladislav Szántó, Automatizácia projektovania integrovaných obvodov ALFA Bratislava. 1985.
2. Alexander Micro Digital logic testing and simulation John Wilay & Sons, 1986
3. INTER Introduction to Inter Cell-Based Design, 1987.
4. National Semiconductor Corporation ASIC Design Manual, 1987.

ABSTRACT

Logic Capture for Logic Simulation on IBM/PC

The article refer to SCHED system, a logic capture for logic simulation. It deals with schematics library organization, draw's scheduler and an algorithm for translating from schema in to netlist.