

PCBD 数据库的建立

李 天 祥

摘要: 本文叙述了 PCBD 中一体化数据库的建立规则及部分实例。

一、前 言

在E-CAD运行中,数据库和数据库的管理是非常重要的。它直接关系到软件的应用范围和效能。专门用来设计印刷电路板(Print Circuit Board Design)以下简称PCBD的应用软件SECMAI,

表1 SECMAI 的元件库(部分)

STANDARD LIBRARIES

```

0
=
Schematic Symbol name...: K
Fan out code.....: 1
Component code.....: 3
Number of functions.....: 2
Reference designator.....: W
Specify value.....: 0
Shape number.....: 6
Function name.....: 0
Number of pins per gate...: 0
Request net name.....: 2
Specify part number.....: 0
Part number.....:
Net name or value.....: B

```

033M50V

=====

```

Schematic Symbol name...: C1
Fan out code.....: 1
Component code.....: 0
Number of functions.....: 1
Reference designator.....: C
Specify value.....: 1
Shape number.....: 603
Function name.....: 0
Number of pins per gate...: 0
Request net name.....: 0
Specify part number.....: 0
Part number.....:
Net name or value.....:

```

```

Schematic Symbol name...: S1
Fan out code.....: 1
Component code.....: 0
Number of functions.....: 1
Reference designator.....: 0
Specify value.....: 0
Shape number.....: 8
Function name.....: 2
Number of pins per gate...: 1
Request net name.....: 5
Specify part number.....: 0

```

原来的数据库,由于是法国标准,与我国国家标准差异较大;又加之原数据库中错误之处时有发生(如表1所示,元件库中O, 033M50V等根本无法使用)。因此,在PCBD中不能使用原数据库,必须重新建立数据库。我们根据国家标准局1986—05—01实施的GB4728.1—85“中华人民共和国国家标准电气图用图形符号”,建立了一个一体化的数据库。

本数据库包含四个方面,即符号库(Symbol Library);元件库(Component Library);功能库(Function Library)形状库(Shape Library)。

二、符号库的建立

在生成数据库的整个过程中,人机对话的交互式方式,以及层次分明的特点,贯彻始终。

符号库的生成共分为9层,它们是:第一层(Level 1):生成符号的“脚”(Pin);第二层(Level 2):生成符号的外形;第三层(Level 3):加注文字说明;第四层(Level 4):加注文字说明;第五层(Level 5):生成固定的Pin号;第六层(Level 6):加注文字说明;第七层(Level 7):生成活动的Pin号;第八层(Level 8):赋予变量;第九层(Level 9):赋予逻辑功能。

生成符号库的具体做法如下:

键入SPCB, 屏幕显示出, CIRCUIT: Creation of a Schematic; LIG SYM: Creation of a Symbol; BOARD: Creation of a Board; SHAPE: Creation of a Shape; Your Choice.....?

选取 LIG SYM, 即生成一个逻辑符号。为简便起见, 可以只键入一个字母 L 即可。

这时屏幕出现: Symbol name? 键入名字, 如 SOO,

屏幕出现: Level number to select? 键入 2, 进入第二层, 以便先生成符号的外形。此时, 屏幕刷新, 出现了一套完整的“菜单”(MENU) 供你选择。当正确地选择菜单 LINE 后, 画出 SOO 的外形如图 1 所示。

再键入 CL (CHANGE LEVEL)。键入 1, 在这一层, 生成 Pin, 如图 2 所示。

键入 CL, 再键入 3, 选取菜单 TEXT, 写 &, 表示该元件是“与非门”, 如图 3 所示。

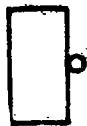


图 1 SOO 的外形

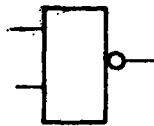


图 2 生成 Pin

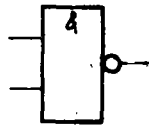


图 3 加注文字说明

再键入 CL, 键入 7, 进入第 7 层。因为 SOO 表示其代表的元件为 74LSOO, 是一个四与非门, 在元件内部, “门”之间可以互换; 在每个“门”中, 输入的两个 Pin 也可以互换。因此, 它的 Pin 是活动的 (Free), 一定要在第 7 层生成如图 4 所示。

再键入 CL, 键入 8, 选取菜单 TEXT, 写出 /51, /52, 如图 5 所示。这一层是赋予符号 SOO 一对变量, /51 和 /52。在实际应用中 (如草图捕获阶段), /52 将变成 74LSOO; /51 将变成 U1, U2, U3……第 8 层完成后, 键入 CL, 再键入 9。进入第九层后, 必须先选取菜单 FIGURES 中的 LEVEL 9, 16, 17, 则屏幕上立即出现: Pin with net name: y/n[n]? 键入 “RETURN”

屏幕又出现: Macro pin: (y/n) [n]? 再次 “RETURN”, 此时, 可以给每个 Pin 赋予, Pin 1, 0, 0, 0; Pin 2, 0, 0, 0; Pin 3, 0, 0, 0。再一次选取 LEVEL 9, 16, 17

屏幕又出现: Pin with net name: (y/n)[n]? 键入 y,

屏幕出现: Signal number (from 1 to 9)? 键入 1, 然后, 在图上取一点, 同时键入 7, 并命名它为 GND (接地符号) 再键入 14, 命名它为 VCC (电源符号) 如图 6 所示。

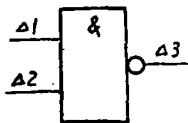


图 4 生成活动 Pin 号

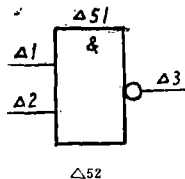


图 5 赋予变量

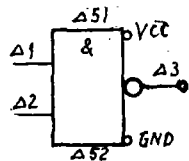


图 6 赋予逻辑功能

最后, 将生成的符号 SOO 存入硬盘中, 在目录中将出现该符号文件为 SOO.DAT.001 值得指出的是, 在第九层中, 顺序是十分重要的。赋予逻辑值的顺序一定是先“活动 Pin”, 后“固定 Pin”。否则, 符号 Pin 的顺序会发生错误。图 7 展示了数据库部分符号。目前已生成常用的 IC、分离元件符号 130 多个。

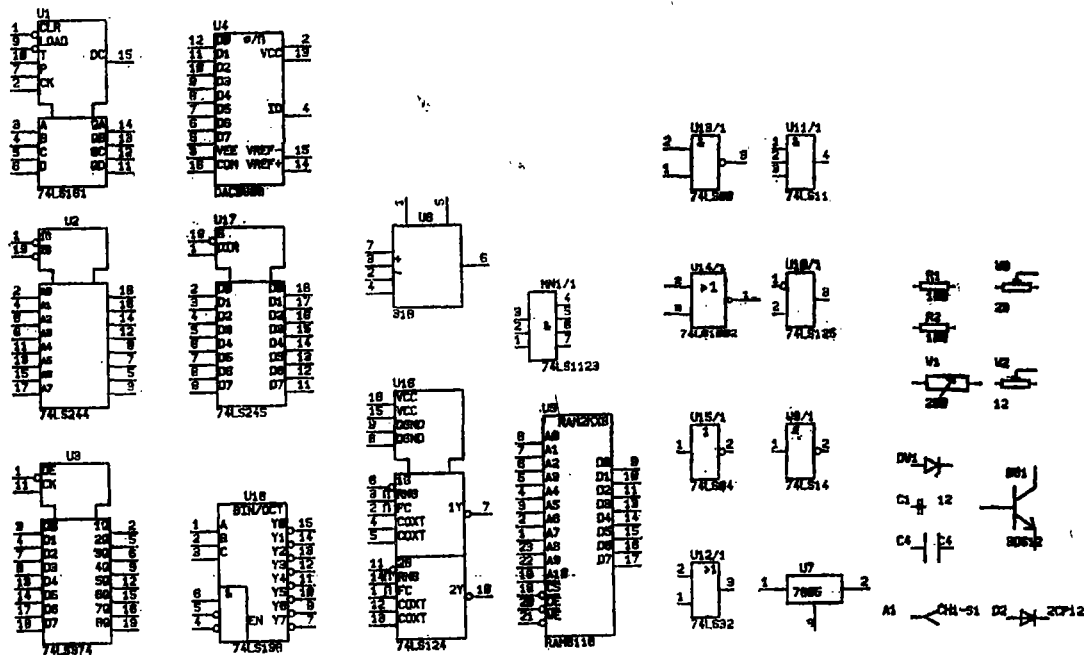


图7 符号库中的部分符号

三、元件库的建立

只生成符号库是不够的，它还不能独立地使用，必须与元件库结合起来，才能变“活”。

元件库包含了元件名、功能类型、形状名、参考符号等等。这里仍以 SOO 为例来阐述元件库的建立过程。

键入 CRECAT，则屏幕出现：CC: Create standard library component. DC: DELETE standard library component. CF: Create standard library function. DE: DELETE standard library function. END: END Program. Your choice……?

选取 CC，表示生成一个元件，屏幕出现：

Component name (14 characters)……? 输入元件名，如 74LSOO，元件的名字不得超过 14 个字符。

屏幕出现：

Part name (6c)……?

键入 SOO，此项名字必须与符号名完全一致，否则出错。Part name 不得超过 6 个字符。

屏幕显示出：

Fan out code (1, 2 or 3)? 通常键入 2，屏幕出现：COMPONENT TYPE, ALWY; FUNC; COIN; TESP; POSY; MULF; PRSY; MACO;

在此例中，应选 FUNC，屏幕出现：Number of functions per package……? 根据 74LSOO 的结构，应当回答“4”（74LSOO 的结构如图 8 所示）

屏幕出现：Reference designator (2c)……? 回答“D”，不得超过 2 个字符。在实际应用中，此项按顺序变为 D1, D2, ……屏幕又显示：

Value to be specified; no = 0, yes = 1, fixed value = 2?

回答“0”

屏幕出现: Shape number.....?这项表示形状号码, 因为74LS00有14个Pin, 所以回答“514”。

屏幕出现: Function name (F,G,P+NO)? 这项表示三种类型的功能器件, 回答“F”, 屏幕又显示: Number of pins per gate? 回答“3”

屏幕又出现: NET TYPE, NO: YES; FIX; VAP; 回答“N”

屏幕出现: Part number to be specified; (y/n) [n].....? Part number (14c.....?)

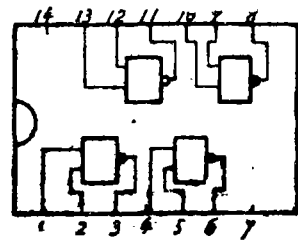


图8 74LS00结构图

这两项可以“RETURN”通过。这样, 元件74LS00就生成了。它可以做为一个独立的元件, 随时调出使用。

模拟元件(ALWY)的生成与74LS00的生成类似, 限于篇幅, 不再赘述。表2列出了两种元件的明细。

表2 生成的元件库(部分)

```

74LS00
=====
Schematic Symbol name...: S1000
Fan out code.....: 2
Component code.....: 1
Number of functions.....: 4
Reference designator...: MM
Specify value.....: 0
Shape number.....: 514
Function name.....: FO
Number of pins per gate.: 3
Request net name.....: 0
Specify part number.....: 0
Part number.....: T174LS00
Net name or value.....:

```

```

74LS04
=====
Schematic Symbol name...: S1004
Fan out code.....: 2
Component code.....: 1
Number of functions.....: 4
Reference designator...: MM
Specify value.....: 0
Shape number.....: 514
Function name.....: F4
Number of pins per gate.: 2
Request net name.....: 0
Specify part number.....: 0
Part number.....: T174LS04
Net name or value.....:

```

四、功能库的建立

在功能库中, 有三种类型元件, 它们是F型、G型、P型。

F型功能元件: 在单个元件中, 有多个逻辑门, 这些“门”可以在元件内部互换, 也可以在同类元件中交换。如74LS00, 74LS04。而有的“门”中的Pin也可以交换, 如74LS00中的Pin1与Pin2; Pin4与Pin5...就可以交换。

G型功能元件: 在单个元件中, 有多重的同一功能, 但整个符号不可分。在元件内部, 同类功能及Pin可以交换。如G153(74LS153)

P型元件: 在单个元件中, 有等效的Pin, 在元件内部可以互换。如74LS30 (P30)。

三种元件的结构示于图9中。

(a) 为74LS00, 可表示为FO; (b)为74LS153, 可表示为G153; (c)为74LS30, 可表示为P30

功能库的做法与元件库的生成类似, 因篇幅所限, 恕不详叙。表3列出了部分功能元件。

五、形状库的建立

形状库, 也称为实体库。它表达了在PCBD中所用元器件的实实在在的形状大小, 必须

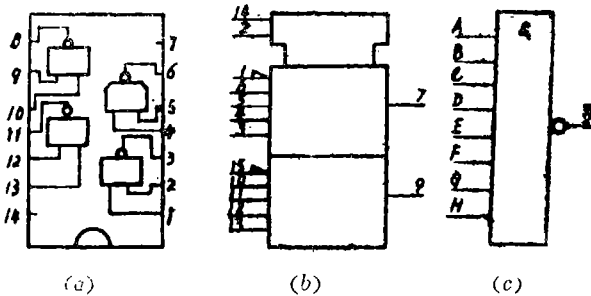


图9 F、G、P类型元件示意图

表3 部分功能元件库明细

FUNCTION F 0

Number of functions per package..... 4
 Number of pins per function..... 3
 Number of functions with swappable pins... 4
 Number of swappable pins per function..... 2

FUNCTION 1 :Pins..... 1, 2, 3
 Coordinates center of gravity : 0, 2
 FUNCTION 2 :Pins..... 4, 5, 6
 Coordinates center of gravity : 0, 3
 FUNCTION 3 :Pins..... 10, 9, 8
 Coordinates center of gravity : -4, 10
 FUNCTION 4 :Pins..... 10, 12, 11
 Coordinates center of gravity : -4, 4

Swappable pins in function 1..... 1, 2
 2... 4, 6
 3... 10, 9
 4... 10, 12

FUNCTION G 873

Number of functions per package..... 8
 Number of pins per function..... 2
 Number of functions with swappable pins... 0
 Number of swappable pins per function..... 0

FUNCTION 1 :Pins..... 8, 9
 Coordinates center of gravity : 9, 9
 FUNCTION 2 :Pins..... 6, 7
 Coordinates center of gravity : 0, 7
 FUNCTION 3 :Pins..... 7, 6
 Coordinates center of gravity : 0, 11
 FUNCTION 4 :Pins..... 8, 9
 Coordinates center of gravity : 0, 12
 FUNCTION 5 :Pins..... 13, 12
 Coordinates center of gravity : -4, 16
 FUNCTION 6 :Pins..... 14, 15
 Coordinates center of gravity : -4, 11
 FUNCTION 7 :Pins..... 17, 16
 Coordinates center of gravity : -4, 7
 FUNCTION 8 :Pins..... 18, 19
 Coordinates center of gravity : -4, 3

FUNCTION P 30

Number of functions per package..... 0
 Number of pins per function..... 0
 Number of functions with swappable pins... 1
 Number of swappable pins per function..... 8

Swappable pins in function 1..... 1, 2, 3, 4, 5, 6, 11, 12

与实物一模一样。否则，即使是设计成功的印刷电路板也会出现元件排列重迭现象。因此，在生成形状库时，尺寸是十分重要的。首先，要生成一个尺寸明细表，以备生成形状库时使用。为此，

键入指令CREDIAPH，

通过人机交互（详细步骤从略），生成 0—47号尺寸表。然后，按着 1—18层的顺序，逐步完成一个实体形状的生成。示意图如10所示。请注意，元器件的生成，一定要从“召唤点”开始，否则就是无效的。图11展示出形状库中的部分“形状”。

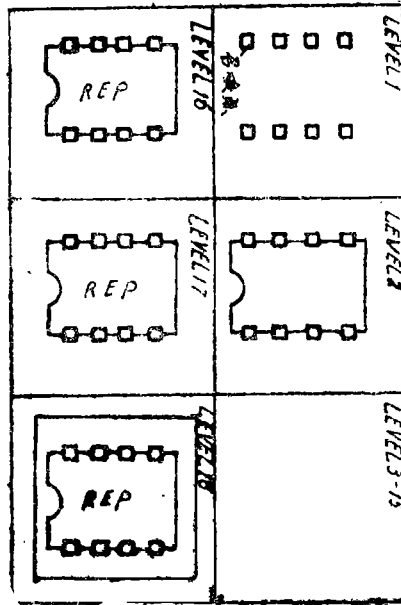


图10 形状生成示意图

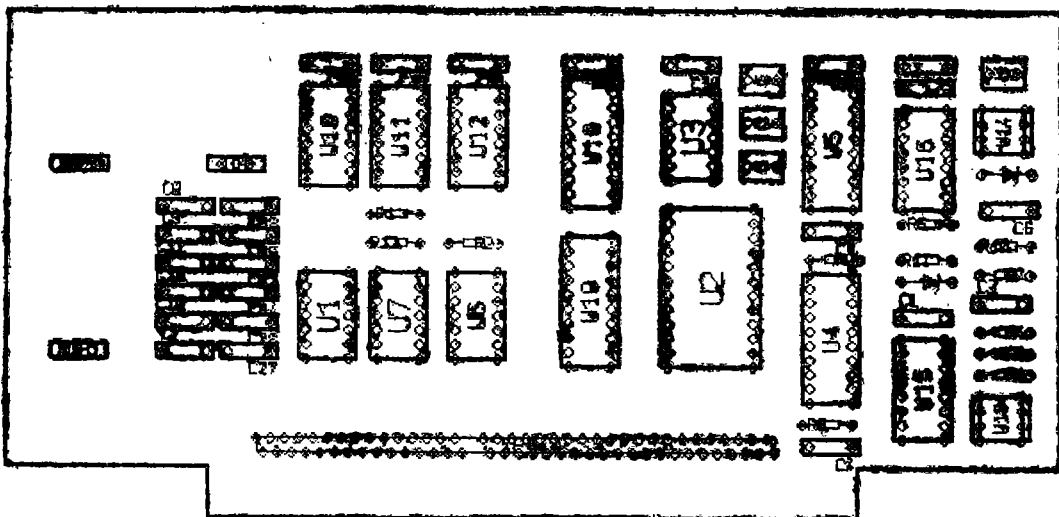


图11 形状库中的部分形状

以上，分别生成了符号库、元件库、功能库和形状库。它们虽然相对地独立，又是密切相关的。元件库中包含了符号、功能号和形状号。因此，也可以说，元件库包含了其他三个库。他们是一体化的数据库，图12表达了它们之间的关系。

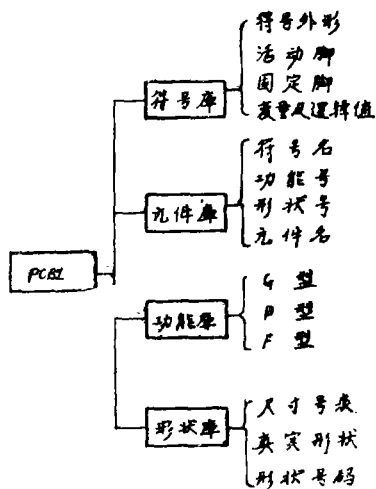


图12 一体化数据库示意图

到目前为止，根据科研生产的需要，我们已建成元件、符号 100 余种，功能器件 20 余种，基本上满足了设计的需要。随着工作的开展，数据库必然会不断被充实。不仅要生成单一的元件，而且还要生成单元电路库，从而使PCBD的速度加快，质量提高。

Creation of PCBD Engineering Databases

Li Tianxiang

Abstract

This paper deals with the creation rule of integrated database in PCBD and some examples are also described.