

# 中华人民共和国国家标准

GB/T 43863—2024

## 大规模集成电路(LSI) 封装 印制电路板共通设计结构

Format for LSI—Package—Board interoperable design

(IEC 63055:2023, MOD)

2024-04-25 发布

2024-08-01 实施

国家市场监督管理总局  
国家标准化管理委员会 发布





目 次

前言 ..... III

1 范围 ..... 1

2 规范性引用文件 ..... 1

3 术语、定义和缩略语..... 1

    3.1 术语和定义 ..... 1

    3.2 缩略语 ..... 3

4 LPB 格式的概念 ..... 4

    4.1 技术背景 ..... 4

    4.2 传统设计 ..... 4

    4.3 传统设计存在的问题 ..... 4

    4.4 LPB 共通设计的概念 ..... 5

    4.5 LPB 共通设计的价值 ..... 5

    4.6 LPB 格式 ..... 6

    4.7 LPB 格式文件概要 ..... 6

5 Basics 语言 ..... 11

    5.1 概述 ..... 11

    5.2 排版和句法 ..... 11

    5.3 字符信息 ..... 11

    5.4 浮点数的表示法 ..... 12

    5.5 文件命名定义 ..... 12

6 M 格式、C 格式、R 格式中的共通语句 ..... 12

    6.1 通则 ..... 12

    6.2 Extensions 语句 ..... 12

    6.3 Header 语句 ..... 13

    6.4 Global 语句 ..... 14

7 M 格式 ..... 28

    7.1 M 格式文件结构..... 28

    7.2 include 语句 ..... 28

    7.3 current\_phase 语句 ..... 29

    7.4 class 语句 ..... 30

8 C 格式 ..... 35

    8.1 C 格式文件结构 ..... 35

    8.2 module 语句 ..... 35

8.3 component 语句 ..... 126

9 R 格式 ..... 130

9.1 R 格式文件结构 ..... 130

9.2 Physicaldesign 语句 ..... 131

9.3 Constrainerule 语句 ..... 163

10 N 格式 ..... 172

10.1 N 格式文件目的 ..... 172

10.2 如何定义网表中的电源层/接地层 ..... 172

10.3 示例 ..... 172

11 G 格式..... 173

11.1 G 格式 Basics 语言 ..... 173

11.2 结构 ..... 173

11.3 Header 部分 ..... 175

11.4 Material 部分 ..... 175

11.5 Layer 部分 ..... 176

11.6 Shape 部分 ..... 176

11.7 Board geometry 部分 ..... 180

11.8 Padstack 部分 ..... 181

11.9 Part 部分 ..... 182

11.10 Component 部分 ..... 184

11.11 Net attribute 部分 ..... 185

11.12 Netlist 部分 ..... 185

11.13 Via 部分 ..... 187

11.14 Bondwire 部分 ..... 188

11.15 Route 部分 ..... 189

# 前 言

本文件按照 GB/T 1.1—2020《标准化工作导则 第 1 部分：标准化文件的结构和起草规则》的规定起草。

本文件修改采用 IEC 63055:2023《大规模集成电路(LSI) 封装 印制板共通设计结构》。

本文件与 IEC 63055:2023 相比做了下述结构调整：

- 删除了 IEC 63055:2023 中 1.4 有关标准内容介绍部分、1.5 词汇使用说明部分、资料性附录 A～附录 H；
- 将 IEC 63055:2023 的 1.3 关键要素介绍部分移至本文件的 4.1 技术背景部分；
- 将 IEC 63055:2023 的 6.4.3 的悬置段改为 6.4.3.1，后续编号顺延；
- 将 IEC 63055:2023 的 6.4.5.4 悬置段删除，原 6.4.5.4.1 改为 6.4.5.4；
- 将 IEC 63055:2023 的 9.2.7 的悬置段改为 9.2.7.1，后续编号顺延。

本文件与 IEC 63055:2023 的技术差异及其原因如下：

- 删除了 IEC 63055:2023 第 3 章中“元件孔(component hole)、球(ball)、电路板(board)、元器件(component)、钻孔(drill)、孔(hole)、连接盘(land)、导线(line 无)、连接盘孔(landless hole)、安装孔(mounting hole)、封装(package)、盘(pad)、管脚(pin)、镀覆孔(plated-through hole)、焊球(solder ball)、导通孔(via)、空洞(void)”的术语和定义，这些术语和定义在 GB/T 2036 和 SJ/T 10668 中有界定。
- 将 IEC 63055:2023 中的“ANSI 标准 X3.4—1986”替换为 GB/T 1988—1998(见 5.3)，以适应我国的技术条件、增加可操作性。
- 将 IEC 63055:2023 中规范性引用的 IEEE Std 1364<sup>TM</sup> 替换为 GB/T 2036、GB/T 14113、SJ/T 10668(见第 3 章)，增加可操作性。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别专利的责任。

本文件由中华人民共和国工业和信息化部提出。

本文件由全国印制电路标准化技术委员会(SAC/TC 47)归口。

本文件起草单位：中国电子科技集团公司第十五研究所、无锡市同步电子科技有限公司、中国航天科工集团第三研究院第八三五八研究所、广东正业科技股份有限公司、中国电子技术标准化研究院。

本文件主要起草人：何安、陈懿、叶伟、徐地华、郭晓宇、拜卫东、范斌、杨鹏、陈长生、楼亚芬、曹易。



# 大规模集成电路(LSI) 封装 印制电路板共通设计结构

## 1 范围

本文件规定了大规模集成电路(LSI)、封装和印制电路板之间共通的设计格式要求。  
本文件适用于大规模集成电路、封装和印制电路板共通设计数据的交换和处理。

## 2 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中,注日期的引用文件,仅该日期对应的版本适用于本文件;不注日期的引用文件,其最新版本(包括所有的修改单)适用于本文件。

- GB/T 1988—1998 信息技术 信息交换用七位编码字符集
- GB/T 2036 印制电路术语
- GB/T 14113 半导体集成电路封装术语
- SJ/T 10668 电子组装技术术语

## 3 术语、定义和缩略语

### 3.1 术语和定义

GB/T 2036、GB/T 14113 和 SJ/T 10668 界定的术语和定义适用于本文件。

#### 3.1.1

**反焊盘 antipad**

导通孔和非连接金属层之间的隔离区,主要用于印制电路板和 LSI 封装。

注:反焊盘的形状主要由印制电路板或 LSI 的工艺极限,以及设计规则格式(R 格式)文件中的焊盘格式来确定。

#### 3.1.2

**球栅阵列封装 ball grid array package**

**BGA 封装 BGA package**

一种表面封装形式,在网格模式中一面覆盖(或部分覆盖),以焊锡球排列。

#### 3.1.3

**键合区 bonding finger**

LSI 封装表面的金属电极,起连接键合丝与内部互连图形的作用。

注:在 LPB 格式文件中,键合区的形状由元器件格式(C 格式)中的焊盘定义。

#### 3.1.4

**键合丝 bonding wire**

连接管芯和键合区的金属丝。

注:在 LPB 格式文件中,键合丝的形状在 R 格式文件中定义。

#### 3.1.5

**时钟 clock**

同步电路中使用的信号。

注：针对寄存器或触发器，所有同步电路都使用时钟信号来同步电路的不同部分。时钟从单个源分配到寄存器或触发器，并同时到达所有这些部件。

3.1.6

延迟 delay

输入信号电平的阶跃函数变化与输出信号传输的大小接近其初始值的瞬间的时间间隔，晶体管的转换时间和信号在线路的传播时间。

3.1.7

管芯 die

裸芯片

晶片的分离部分(或整体)，用于在设备中执行一个或多个功能，在小块半导体材料上制作给定的功能电路。

3.1.8

差分信号 differential signal

用两个互补信号在成对的两条信号线上传送信息的方法。

3.1.9

保护屏蔽 guard shield

为机械保护提供的屏障或外壳，当放置在接地面(GND)导体上时具有接地的功能，目的是限制来自其他信号的电磁干扰。

注：也称为“接地屏蔽”。

3.1.10

网络 net

代表电气连接中理想元件的相对位置或端子互连之间的标记。

3.1.11

封装塑模 package mold

用树脂保护 LSI 芯片免受应力、外力、水、静电和外来物质的影响。

注：封装塑模包含树脂、二氧化硅、碳和阻燃材料。

3.1.12

焊盘堆叠 padstack

构成盘各层的组合。

3.1.13

物理设计规则 physical design rule

由半导体制造商提供的一系列参数，使设计人员能够验证掩模组的正确性。设计规则集规定了某些几何和连通性限制，以确保有足够的余量来考虑半导体制造工艺的可变性，从而保证大多数零件能正常工作。

3.1.14

端口 port

提供接入和接收电磁能量、信号的终端或网络，以观察或测量终端或网络设备参数。

3.1.15

电源域 power domain

为了电源管理的目的而作为一个组处理的实例的集合。

注：电源域的实例通常(但并非总是)共享一个主电源集。一个电源域也可能有额外的供应，包括保持和隔离供应。

3.1.16

接收器 receiver

用来进行解释和执行接收信号的装置。

3.1.17

**参考点 reference point**  
用于表示坐标的参照点。

3.1.18

**驱动器 driver**  
产生和发出信号的装置。

3.1.19

**偏移 skew**  
由于信号的传播而引起的延迟时间的变化量,或参考信号与目标信号之间延迟时间的间隔量。

3.1.20

**子电路 sub-circuit**  
特定的一个电路单元。

3.1.21

**终端 terminator**  
安装在电缆末端,确保与系统其他部分的电气连接,并将绝缘保持到连接点的装置。

3.2 缩略语

下列缩略语适用于本文件。

- BGA:球栅阵列(Ball Grid Array)
- DDR:双倍数据速率(Double Data Rate)
- DXF:绘图交换格式(Drawing Exchange Format)
- EDA:电子设计自动化(Electronic Design Automation)
- GDS(GDS II):图形数据库系统(Graphic Database System)
- GND:接地(Ground)
- ICEM:集成电路发射模型(Integrated Circuit Emission Model)
- ICIM:集成电路抗扰度模型(Integrated Circuit Immunity Model)
- I/O:输入输出(Input Output)
- IBIS:输入/输出缓冲区信息规范(Input/output Buffer Information Specification)
- JEITA:日本电子和信息技术工业协会(Japan Electronics and Information Technology Industries Association)
- LPB:大规模集成电路—封装—印制电路板(LSI,Package,and Board;LSI—Package—Board)
- LSI:大规模集成电路(Large-Scale Integration)
- MD5:消息摘要算法 5(Message Digest Algorithm 5)
- PCIE:高速串行双通道高带宽传输(Peripheral Component Interconnect Express)
- PKG:封装(Package)
- PWB:印制线路板(Printed Wiring Board)
- RMS:均方根(Root Mean Square)
- SMA:超小型版 A(subminiature version A)
- SPICE:以集成电路为重点的仿真程序(Simulation Program with Integrated Circuit Emphasis)
- VHDL:超高速集成电路硬件描述语言(Very High Speed Integrated Circuits Hardware Description Language)
- XML:可扩展标记语言(Extensible Markup Language)
- XTAL:晶体(crystal)

4 LPB 格式的概念

4.1 技术背景

由于系统的速度越来越快,接口和电源供应的电压越来越低,使得时钟和噪声的设计余量不断降低,同时,设计中考虑成本和性能的平衡也变得越来越重要。在传统设计中,LSI、封装和印制电路板是按照单独设计准则设计的,然而,随着设计余量的降低,很难分别为每个 LPB 零件提供设计指导。因此,最终的设计目标需要 LPB 各部分设计人员的配合,也就是说,在系统设计过程中,需要创新运用仿真技术来确定设计准则,为了完成这一任务,需要一个快速、准确的仿真环境。

LPB 格式包含以下五类(见图 1):

- a) 项目管理(M 格式);
- b) 网表(N 格式);
- c) 元器件(C 格式);
- d) 设计规则(R 格式);
- e) 结构(G 格式)。

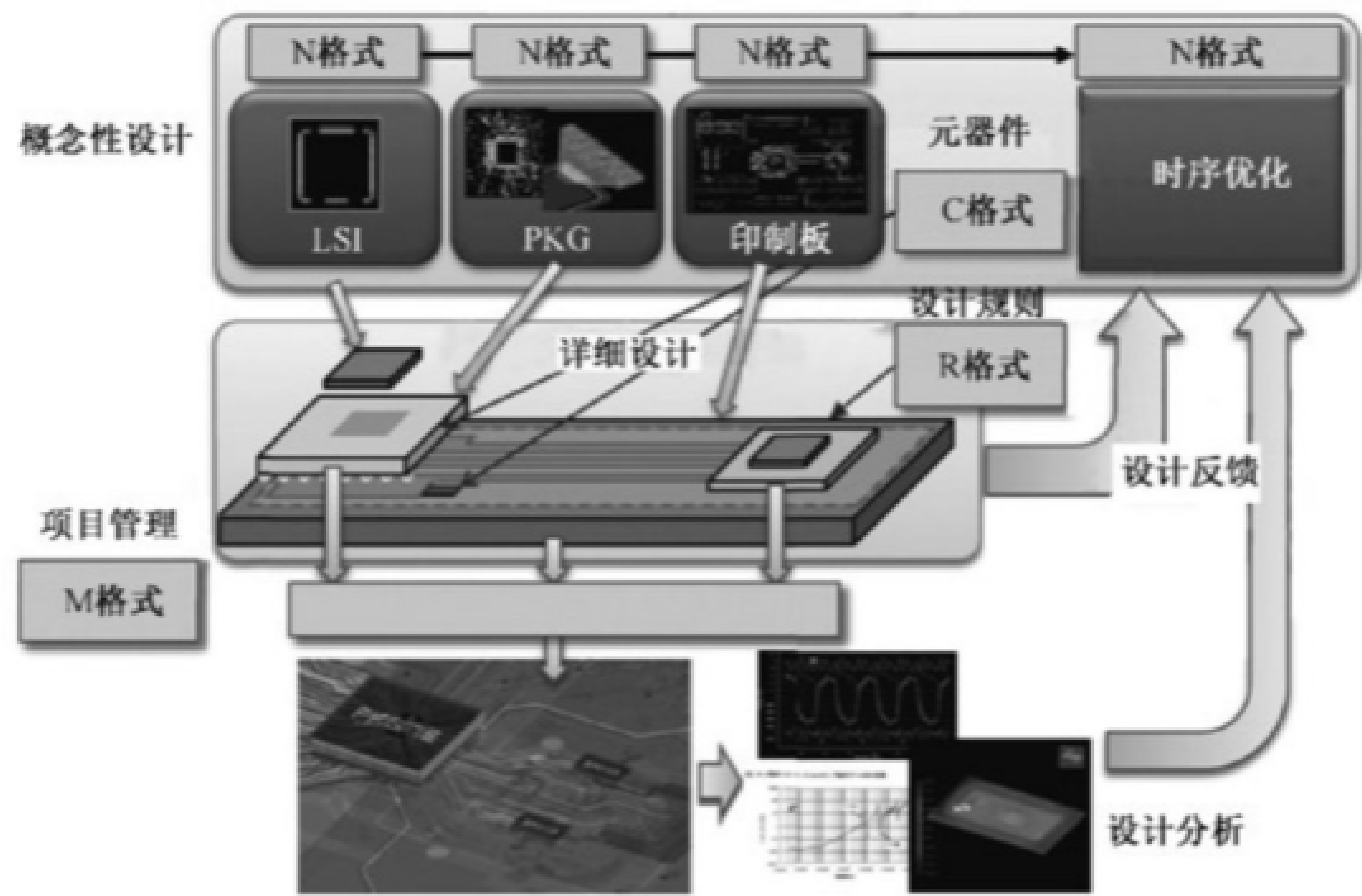


图 1 LPB 格式

4.2 传统设计

在传统设计中,虽然 LPB 各部分对设计准则进行了设计分析,但也只在最初的设计阶段对设计作了部分优化,并未对整个系统设计进行优化和分析,缺乏对整体系统设计的综合考量,未对个别设计点进行合适的设计调整,且在最初的设计阶段对设计余量估计过度。这些都对最终产品的质量和成本产生了显著的影响。

4.3 传统设计存在的问题

4.3.1 设计人员之间存在误解

不同的设计人员认为同一个词有不同的含义;设计人员经常用不同的词来描述相同的现象;设计人员有不同的主观偏好,例如偏好从顶层开始或偏好从底层开始等不同视角;即使设计人员在同一个办公室,也会发生这些情况。



4.3.2 设计信息不足

在设计单个组件时,不能假定 LSI、封装和印制电路板之间的限制和设计余量。设计人员在连接单个组件后可能会遇到很大的障碍,因为他们不确定其他组件的局限性和余量。例如,由于缺乏有关 LSI 焊盘分配、封装焊点分配,以及应在印制电路板上占据的位置的相关信息,会导致设计封装或印制电路板(布线)变得非常困难。

4.3.3 浪费设计时间

由于电子设计自动化(EDA)供应商的输入和输出格式不同,在某些情况下,设计人员需要对不同的 EDA 提交的数据进行格式转换。当设计人员在格式转换时缺少信息并在以后弥补不足时,就会出现问 题。此外,每个 LSI 电路、封装和印制电路板的引脚名和网表名可能不同,设计人员需要调整这些数据,并花费更长的时间进行修正。

4.4 LPB 共通设计的概念

共通设计是针对采用传统 LPB 单独设计方法遇到的问题的一种设计解决方法,它被定义为可被每一个 LPB 分部分使用而实现交互设计的一种设计类型。首先,设计人员采用共通的设计方法,通过对整个 LPB 的设计考虑,制定出满足产品性能标准的最小化设计准则。此后,设计人员根据设计指南对每个部分进行单独设计,在每个单独的设计完成后,设计人员将每个单独部分集成起来进行仿真,确认所有产品的性能,并为所有其他部件提供有关设计的反馈。

4.5 LPB 共通设计的价值

4.5.1 LPB 共通设计的影响

LPB 共通设计是一种在设计 LPB 组件间确保信号完整性和功率完整性的合适方法,该技术是 4.3 中提出或归结的问题的解决方案,使用 LPB 共通设计可以获得减少设计成本和时间 的效果。

4.5.2 通过无迭代的设计流程降低开发成本和时间

当设计人员最终将部分优化设计的零件连接起来时,有时会出现意想不到的设计余量不足,此时,设计人员需要找到 LPB 的哪一部分造成了边际瓶颈。因此,设计人员不仅需要花费大量的时间,还需要额外花费成本来创建所需的设计余量。在这种情况下,在启动 LSI 设计、封装设计和印制板设计时,通过设计人员之间共享信息和协作来减少开发成本和时间。图 2 比较了传统的设计流程和 LPB 共通的设计流程,LPB 共通的流程减少了开发成本和时间。

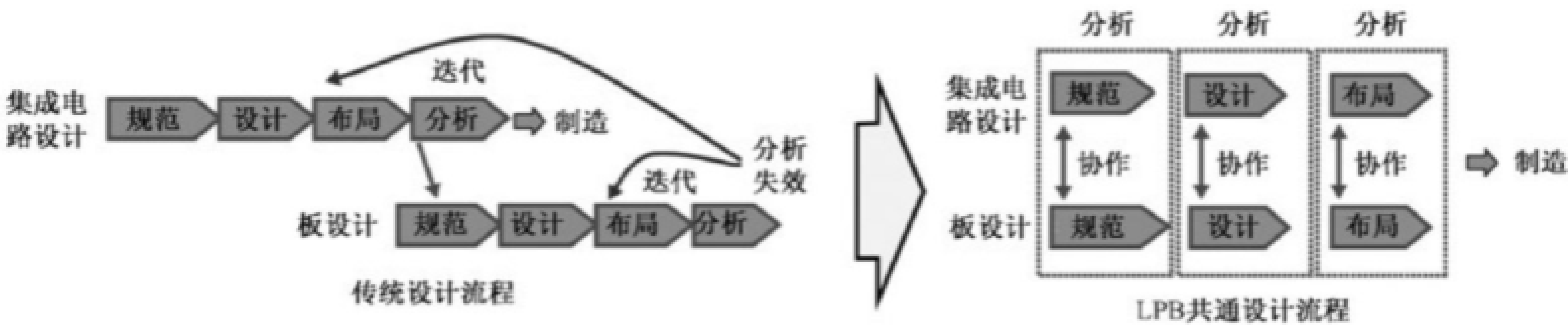


图 2 传统的设计流程和 LPB 共通的设计流程

4.5.3 降低系统设计的总成本

在传统设计中,由于设计人员以单独的余量分别设计 LSI、封装和印制电路板的每个部分,因此当

最终连接这些部件时,整个系统的设计余量有时会变得过大。而利用 LPB 共通设计降低总成本是可能的,因为整个设计的余量被控制和保持在一起,包括减少设计中使用的部件的数量和成本。图 3 比较了传统设计流程和 LPB 共通设计流程的设计余量。

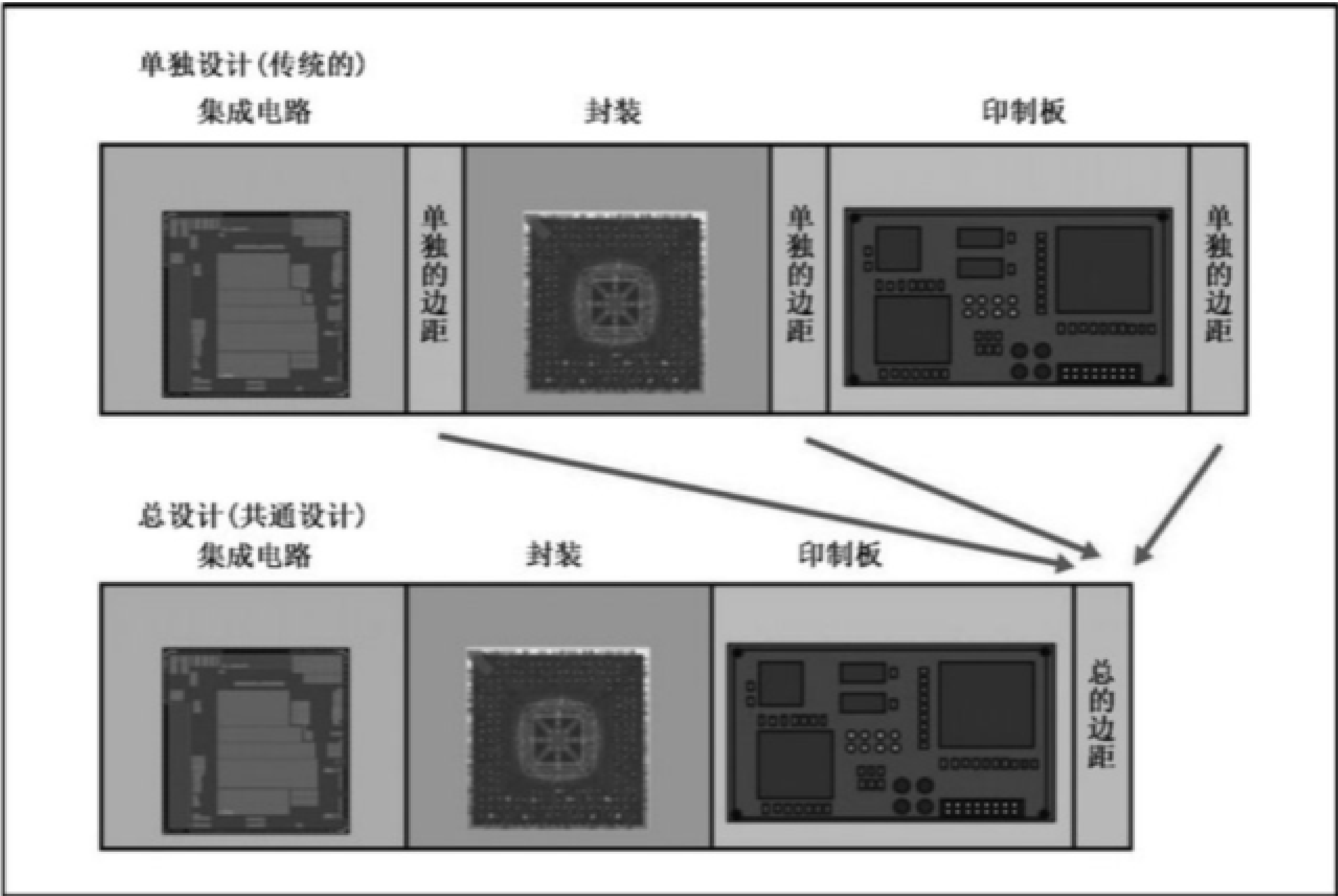


图 3 LPB 共通设计可控制设计余量

4.6 LPB 格式

LPB 格式是描述设计和验证所需信息的通用语言,LPB 格式解决了传统设计方法中存在的问题,提高了产品质量,缩短了设计时间,减少了设计团队开发 EDA 工具的时间;此外,LPB 可以实现设计团队之外的信息的共享,并成为供应链中信息分发的媒介,避免了对设计信息的误解,改善了整个行业的信息流。因此,质量、成本和交货时间(QCD)都将得到改善。

4.7 LPB 格式文件概要

4.7.1 概述

图 4 展示了如何使用 LPB 格式交换设计信息的一个示例,在本例中,存在三种类型的设计工具:LSI 设计工具、封装设计工具和印制板设计工具。所有设计人员都使用 LPB 格式的标准文件交换设计信息,通过统一使用交换设计信息的文件符号用以防止误解,并使设计工具设定自动化。

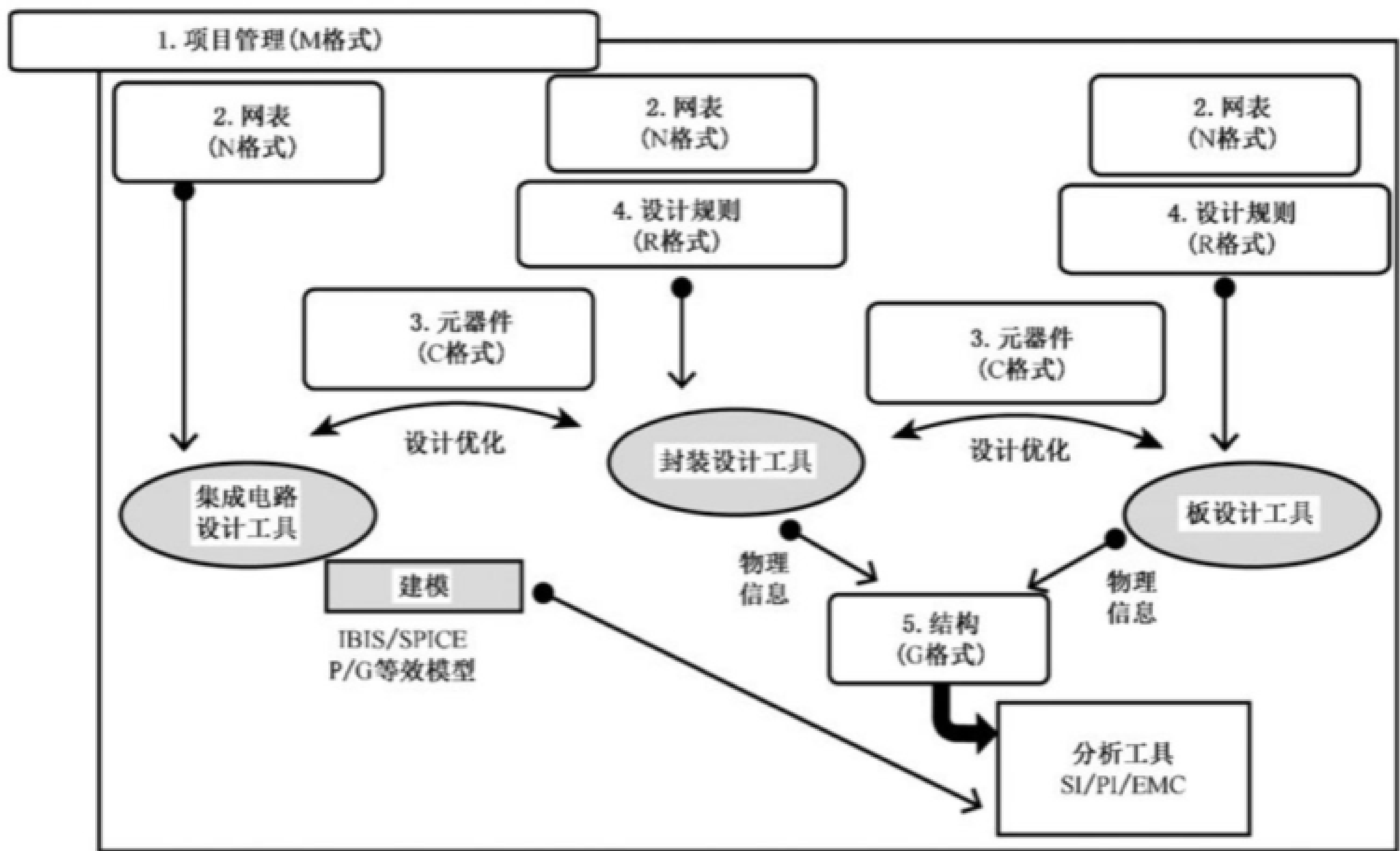


图 4 使用 LPB 格式交换设计信息的示例

4.7.2 C 格式

C 格式的第一个目的是对部件的外部规范(如 LSI、封装、端口等)给出统一的表示法。外部规范是使用这些部件所需的信息,如下所示:

- a) 部件的物理形状;
- b) 部件输入和输出信号的名称和类型;
- c) 输入/输出(I/O)规范,如引脚的物理形状和位置,或可互换的引脚定义;
- d) 设计约束,例如延迟或偏离的上限;
- e) 设计规范,如引脚的输入阻抗或功耗。

通过 C 格式统一的表示法,就可在不产生误解的情况下交换信息。例如,印制电路板设计人员应理解从几个 LSI 供应商交付的规范,并且应建立基于这种理解的设计工具。如果使用每个供应商自己的表示法来描述规范,则在此过程中可能会发生人为错误,使用 C 格式代替供应商自己的符号,可自动设置设计工具,防止因误解而造成人为错误。图 5 显示了使用 C 格式的 LSI 规范信息流的示例,LSI 规范都是由 C 格式提供的,即使供应商不同,印制板设计人员只需阅读一种表示法就能理解规范,而设计工具可通过输入 C 格式实现自动设置。



图 5 使用 C 格式的 LSI 规范信息流示例

统一的表示法也能顺利地传递有关规范更改的信息。图 6 显示了一个共通的设计流程的示例,一个印制板设计人员和一位 LSI 设计人员相互合作。LSI 封装的引脚分配的信息是由 LSI 设计人员提供

的,当印制板设计人员希望更改 LSI 包的引脚分配时,可修改提供的 C 格式文件并将其返回给 LSI 设计人员。使用 C 格式可防止对规范更改的误解,并防止因规范更改而可能发生的人为错误。

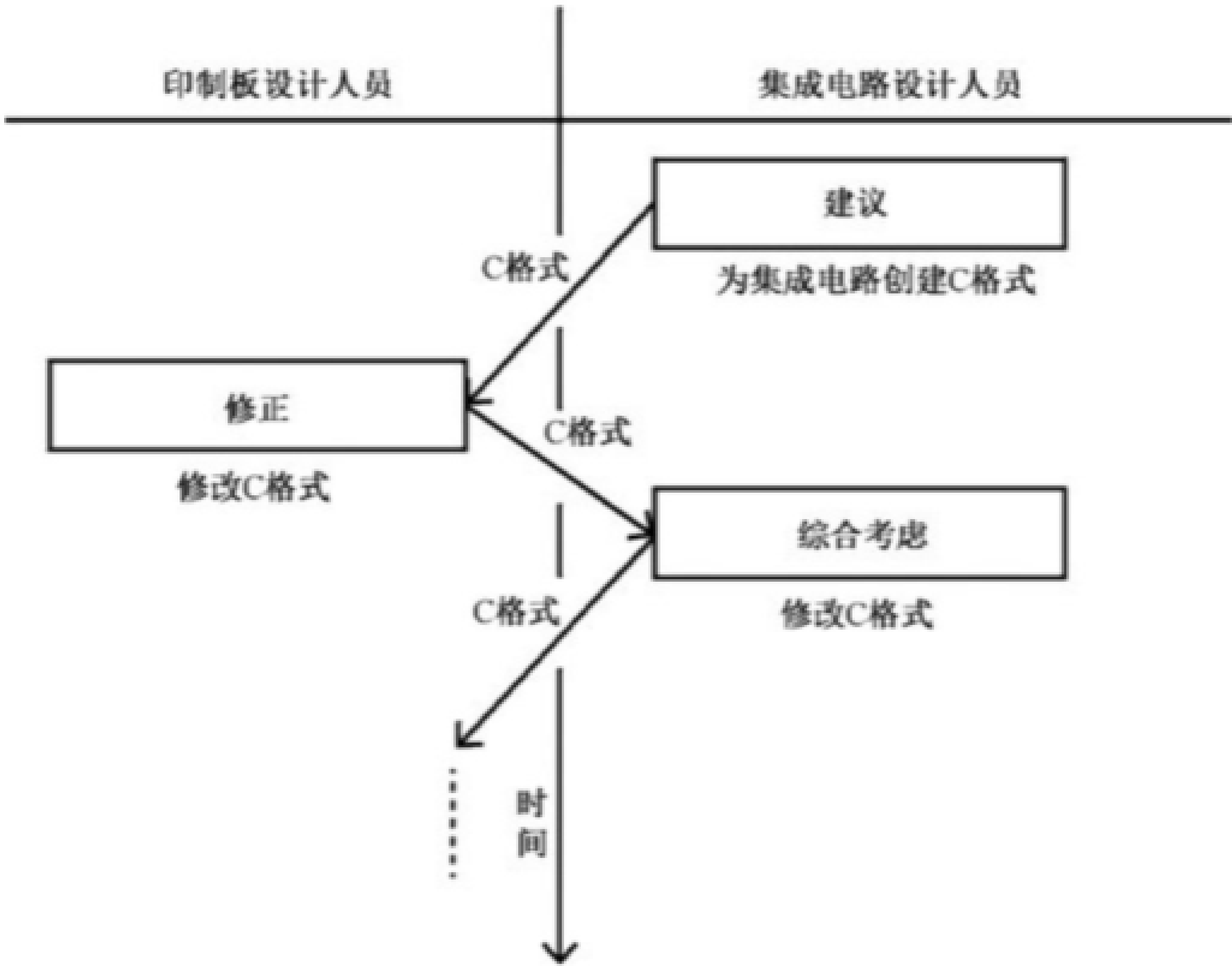


图 6 具有 C 格式的共通设计流程示例

C 格式的第二个目的是为仿真模型提供统一的接口,如仿真程序(SPICE)的网络列表、输入/输出缓冲区信息规范(IBIS、IBIS-AMI)、S-参数等。C 格式包含了这些仿真模型文件,并给出了仿真模型节点与部件物理端口之间的交叉引用方式,通过输入这些模型文件,可将仿真工具自动插入到仿真模型中。图 7 显示了一个交换仿真模型的示例,模型文件与 C 格式文件一起提供。

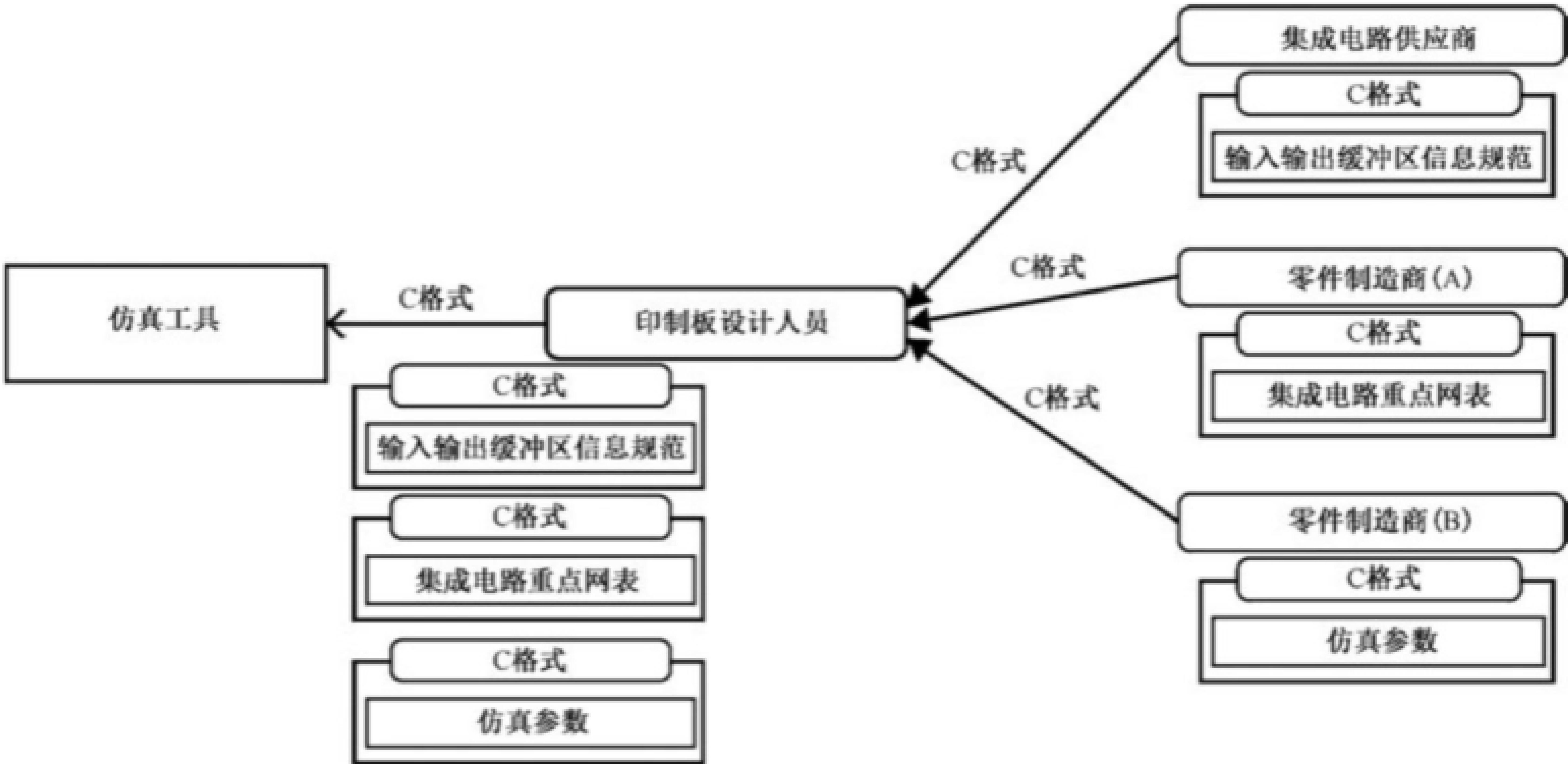


图 7 用 C 格式交换仿真模型的示例

C 格式的第三个目的是提供连接点来合并多个 C 格式封装布局数据文件,如图形数据库系统(GDS)Ⅱ,并在 C 格式文件中的端口和布局数据文件中的对象之间提供交叉引用。例如,当 C 格式文件包含 GDS Ⅱ文件时,C 格式的端口与 GDS Ⅱ文件中的坐标/层相关联;当 C 格式文件包含 G 格式文件时,C 格式文件中的端口与 G 格式文件中的引脚相关联。此功能可用于合并不同设计人员设计的两个或多个布局数据,而当此情况发生时,印制板设计人员可利用封装设计人员提供的封装布局数据来分析印制电路板的信息,并输入到相应的分析工具中。分析工具能使用 C 格式文件中定义的交叉引用找

到连接点,并将印制电路板和封装的版图数据合并。

C 格式的第四个目的是支持印制电路板和封装的布局,格式包括有关零件的放置信息,此功能可用于从系统设计人员到印制板设计人员之间传递布局规划信息。图 8 显示了系统设计人员请求电路板设计人员进行印制板设计时的设计流程示例。在工作开始时,印制板设计人员可根据所提供的 C 格式文件建立设计工具,通过统一平面图的表示法,在不产生误解的情况下交换平面图信息,并自动建立设计工具。

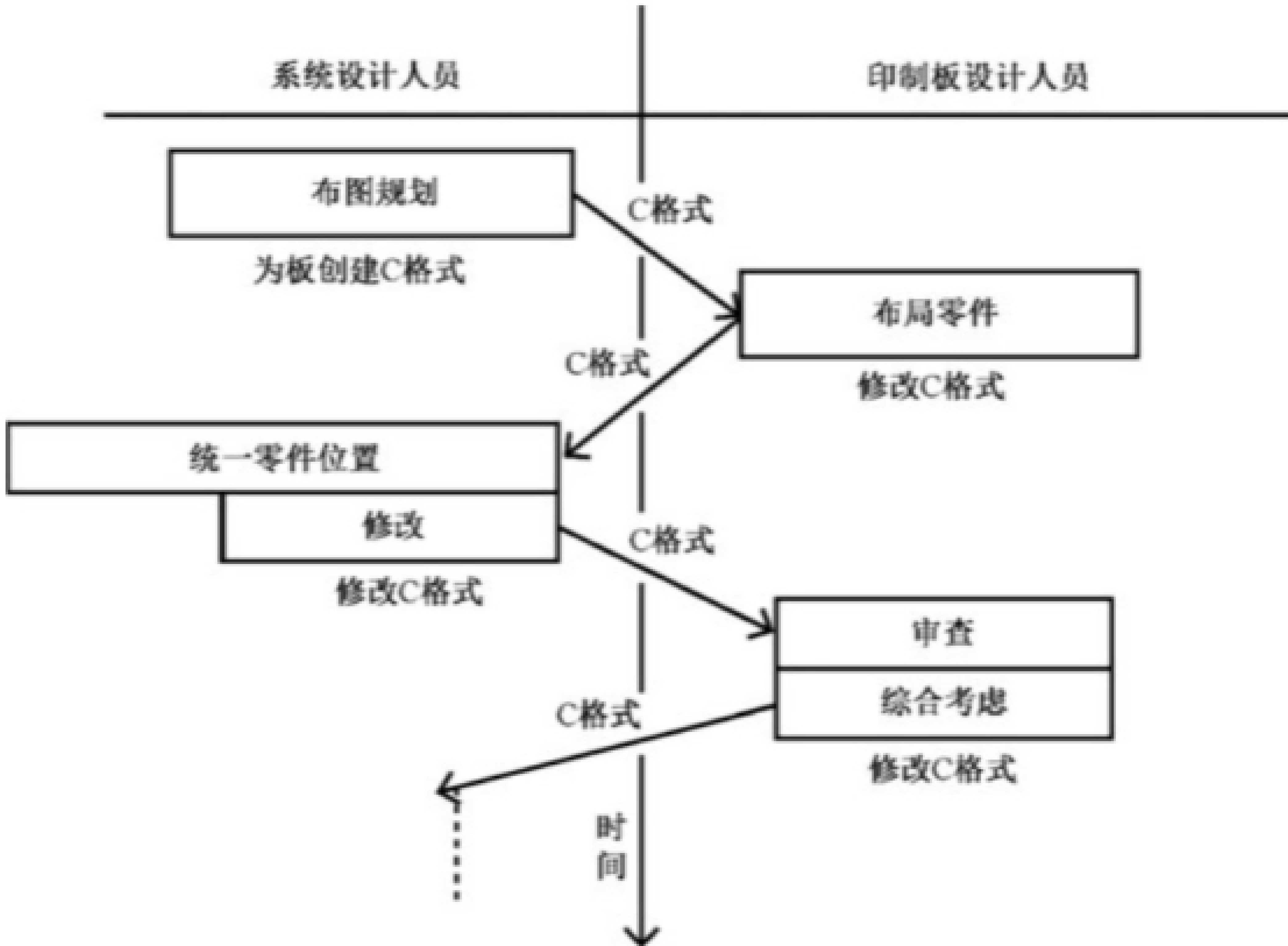


图 8 系统设计人员与印制板设计人员之间的设计流程示例

4.7.3 R 格式

R 格式的第一个目的是统一印制电路板和封装设计规则的符号。以下设计规则以 R 格式定义：

- a) 封装和印制电路板的叠层；
- b) 导电层/绝缘层的厚度；
- c) 每一层所用的材料；
- d) 材料参数,如电导率、介电常数或损耗角正切；
- e) 线宽和线间距；
- f) 导通孔间距；
- g) 孔的类型。

通常,印制电路板和封装的制造商使用它们自己的符号来提供设计规则,设计人员应理解几个符号中描述的设计规则,并设置设计工具,否则有可能出现人为错误的风险。然而,使用统一的符号(如 R 格式),可自动设置设计工具并防止误解。图 9 显示了使用 R 格式交换设计规则的示例,所有制造商提供统一的 R 格式符号设计规则,设计工具可通过输入 R 格式自动设置。



图 9 R 格式交换设计规则的示例

R 格式的第二个目的是定义印制电路板和封装的物理设计约束,其中,物理设计约束意味着安装部件的高度限制和非默认设计规则区域。

4.7.4 G 格式

G 格式的目的在于统一印制电路板或封装叠层结构的布局数据符号。通过布局数据的统一表示法,可在分析工具和布局工具之间无缝地交换布局数据。G 格式文件包括以下几个信息:

- a) 印制电路板;
- b) 材料的叠层和物理参数;
- c) 安装部件的形状和位置;
- d) 管脚的形状和位置;
- e) 网络的走线或图形;
- f) 导通孔的形状和位置;
- g) 键合丝的形状。

图 10 显示了使用 G 格式交换布局数据的示例,如果所有布局工具输出 G 格式文件,则只需输入 G 格式即可设置分析工具。

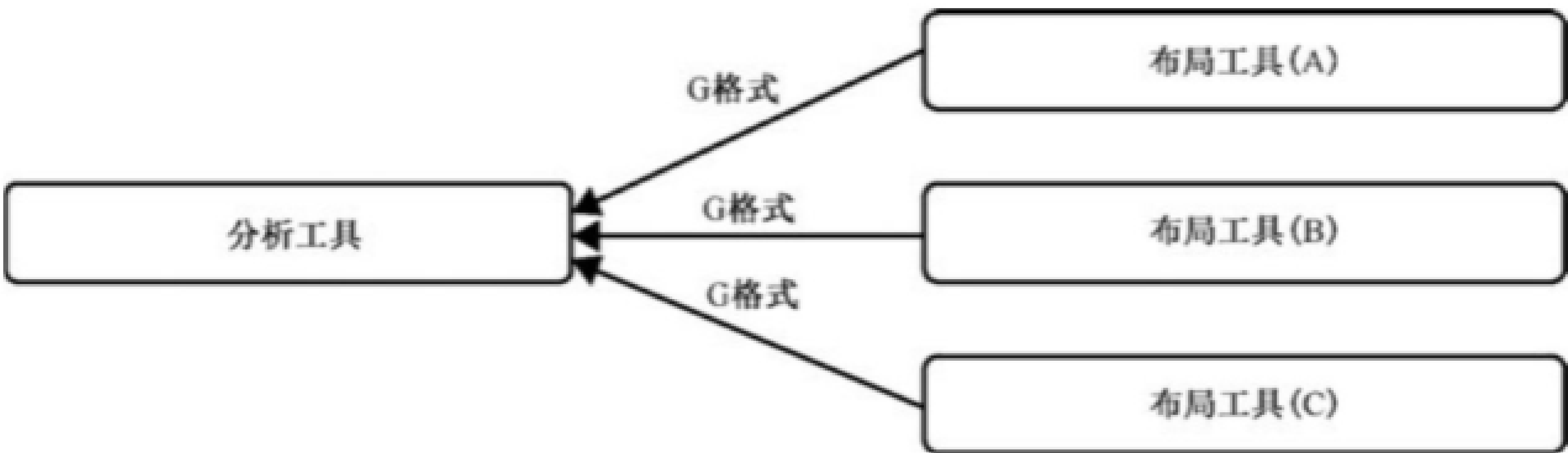


图 10 G 格式交换布局数据的示例

4.7.5 N 格式

N 格式的目的在于统一用于设计印制电路板和封装的网表符号,其中“网表”是指安装在印制电路板或封装上的部件之间的连接信息。N 格式符合硬件描述语言(HDL),并增加了识别电源和接地的关键字。统一的网表符号使得在电路设计工具和布局工具之间能够无缝地交换网表。例如,当电路设计人员向电路印制板设计人员采购印制电路板设计时,设计人员之间应交换网表。

如果网表使用多个符号表示,那么在设计工具中输入部件之间的连接时可能会发生人为错误。然而,通过使用网表的统一表示法,可防止在设置设计工具时出现人为错误,因为所有设计工具都可通过只支持网表文件的一种表示法来设置部件之间的连接。图 11 显示了一个使用 N 格式文件的设计流程示例,电路修改采用 N 格式进行交换。

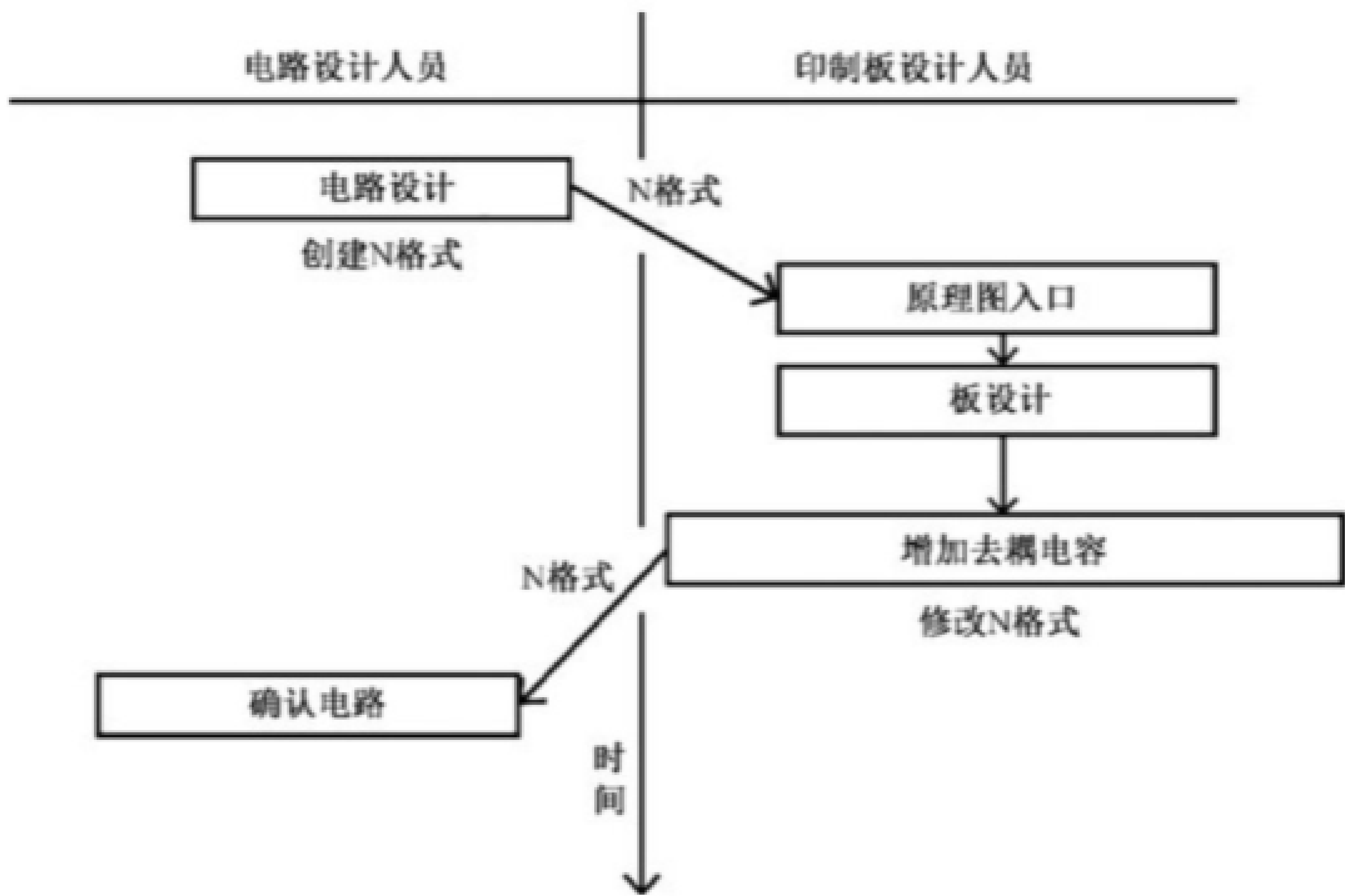


图 11 N 格式设计流程示例

4.7.6 M 格式

LPB 格式文件根据设计进度不断更新，M 格式的目的在于管理每个文件的版本，以防止在交换 LPB 格式文件时出现错误。

5 Basics 语言

5.1 概述

本节描述了 LPB M 格式、C 格式和 R 格式的语法定义中使用的约定。

5.2 排版和句法

下面的列表描述了语法约定：

- a) 文本：单空间字体用于表示应按字面顺序键入的属性或语句；
- b) (印刷字体)斜体的：斜体用于指示用户定义的信息，该信息将被替换为名称或值；
- c) |：垂直条分隔单个属性或语句的可能选择，它们优先于任何其他字符；
- d) []：括号表示可选属性或语句，当与垂直条一起使用时，它们包含一个选项列表；
- e) [...]...：括号后面的三个点表示应指定零或多个属性或语句，当与垂直条一起使用时，它们包含一个选项列表；
- f) { }...：括号后面跟着三个点，表示应指定一个或多个属性或语句，当与垂直条一起使用时，它们包含一个选项列表；
- g) ...：三个点表示可重复先前的值。

M 格式、C 格式和 R 格式中的所有字符串都是区分大小写的。

注：本文件中的所有代码示例都是用单空格字体编写的。

5.3 字符信息

除非另有规定，LPB 应在格式文件中只使用 GB/T 1988—1998 中定义的 ASC II 字符，不准许使用代码大于十六进制 0x7E 的字符，而且不准许使用 ASC II 控制字符(数字小于十六进制 0x20)，但制表符或行终止序列除外。



5.4 浮点数的表示法

以下格式用于表示浮点数。

[**−** | **+**]**<**mantissa**>**[**{**e | E**>**[**−** | **+**]**<**exponent**>**]

**<**mantissa**>**

整数或分数,只能使用数字字符(0 到 9)和十进制分隔符(句点)。

**<**exponent**>**

指数只能使用整数(10 的幂),不应包含十进制分隔符。

5.5 文件命名定义

以下定义了文件引用的规则。

- 目录:在分层文件系统中存储文件的位置。
- 根目录:文件系统根目录的标准名称。在 Windows 中,它包括驱动程序字母(C 行:),在 Linux 中,它包含根斜杠(/)。
- 当前目录:保存当前 LPB 格式文件的目录。
- 路径:指向特定文件的路由,该文件由一系列根目录(可选)和目录表示。
- 绝对路径:绝对路径是从文件系统的根目录开始的路径,它定位指定的文件,而不考虑当前目录。
- 相对路径:相对路径不是从根目录开始的,它定位相对于当前目录的文件。
- 文件引用:对文件的引用,表示为简单的文件名或包含相对或绝对路径。简单文件名表示文件位于当前目录中。

6 M 格式、C 格式、R 格式中的共通语句

6.1 通则

本节提供 M 格式、C 格式和 R 格式的常用语句。

M 格式、C 格式、R 格式使用万维网标准为 XML 1.0 版本,由数个语句构成,语句的名称用三角括号括起来**<**。

**<**element\_name**>**

一个语句可包含多个语句作为其子语句。如果一个语句包含有子语句,那么该语句以**<**element\_name**>**开始,以**</element\_name>**结束。

**<**element\_name**>**.....**</element\_name>**

如果一个语句不包含子语句,则该语句仅表示为**<element\_name/ >**,语句顺序根据 M 格式、C 格式、R 格式的语法规则确定。

语句可有一种或更多属性,属性由名称和用双引号括起来的值表示,属性值可以是数值或字符串,数值应按照 5.4 中定义的规则进行标注,除非受关键字限制,否则字符串值为自由字符串。

attribute\_name=“attribute\_value”

属性能以任何顺序出现,并且应是唯一的。

6.2 Extensions 语句

6.2.1 通则

**<**extensions**>**语句是用户自己的扩展库。它用于将用户自己的规范添加到 M 格式、C 格式和 R 格



式中。所有用户扩展都是使用“命名空间”(XLM 1.0[B16]中的“命名空间”)定义的。任何描述都能添加到用户扩展区域,只要它们符合 XML 格式即可。用户扩展声明具有以下语法:

```
<extensions xmlns:user_prefix="URI">
    {<user_prefix:user_own_tag {user_own_attribute="any_string"}.../>}...
</ extensions >
```

如果多个用户单独添加他们自己的规范,关键字可能会相互冲突。使用 URI 引用标识的“命名空间”能防止关键字冲突。URI 最重要的一点是它是唯一的。需要扩展的用户可通过向互联网域名管理组织注册 URI 中包含的域名来保证 URI 的唯一性。用户扩展名负责保证在域名后使用的字符串的唯一性,LPB 格式会将 URI 用作标识符,包含一个特定的域名是确保 URI 字符串唯一性的方法。这里指定的 URI 可能是虚构的,但应是唯一的。URI 只能使用 ANSI 标准 X3.4—1986(代码 0x21 到 0x7E)中的图形字符。

6.2.2 例句

```
以下是<extensions>语句的一个示例。
<header project="test project " design_revision="1.1">
    <extensions xmlns:A="http://www.A.com">
        <A:own_user_area1myattribute1="xxx" myattribute2="yyy"/>
        <A:own_user_area2myattribute3="xxx" myattribute4="yyy"/>
    </ extensions >
    <extensions xmlns:B="http://www.B.com">
        <B:own_user_area1myattribute1="xxx" myattribute2="yyy"/>
        <B:own_user_area2myattribute3="xxx" myattribute4="yyy"/>
    </ extensions >
</ header >
```

6.3 Header 语句

6.3.1 概述

<Header>语句定义 M 格式、R 格式和 C 格式文件的管理信息。

```
<header
    project="project_name "
    design_revision="revision_number "
    [date="date "]
    [author="owner_of_this_document "]
    [email="email_address "]
    [company="company_name "]
    [comment="any_comment "]
>
    [<extensions> element]...
</header>
```

通过处理〈Header〉语句,能获得项目名称(project)、创建日期(date)、关于作者的信息(author, email, 和 company),以及一个修订编号(design\_revision)。

6.3.2 定义属性

〈Header〉语句的属性定义如下。

- a) project:此属性指定项目的名称,项目中的所有 LPB 格式文件应具有相同的项目名称。
- b) design\_revision:此属性指定每个 LPB 格式文件的修订编号,修改 LPB 格式文件时,应适当增加修订编号。
- c) permission:此属性指定每个 LPB 格式文件的编辑权限。该值应为下列值之一:  
EDITABLE 允许编辑;  
LOCKED 禁止编辑。
- d) date:此属性指定 LPB 格式文件的创建或修改日期。
- e) author:此属性指定 LPB 格式文件的作者的名称。
- f) email:此属性指定作者的电子邮件地址。
- g) company:此属性指定此设计文件的作者所属的组织。
- h) comment:此属性指定用作注释的任意文本字符串。

6.3.3 语句组成

〈header〉语句包含以下内容:  
〈extensions〉

6.3.4 例句

以下是〈header〉语句的一个示例。

```
〈header
  project="JEITA_LPB_SAMPLE_PROJECT"
  design_revision="1.3"
  date="20120331"
  permission="LOCKED"
  author="yyyyy xxxxx"
  email="xxxx@jeita.jp"
  company="JEITA"
  comment="This is a sample code of LPB format."
〉
  〈extensions xmlns:LPB_FORUM=http://www.lpb-forum.com〉
    〈 LPB_FORUM:header_ext comment= "this is a sample of user extension area" /〉
  〈/extensions〉
〈/header〉
```

6.4 Global 语句

6.4.1 概述

〈global〉语句定义了 在 R 格式和 C 格式文件中使用的单元系统、基本形状和焊盘堆叠。

```
<global>
    <unit> element
    [<shape> element]...
    [<padstack_def> element]...
    [<extensions> element]...
</global>
```

定义变量的范围仅限于声明该变量的文件。<global>语句的内容由一个<unit>语句和一个或零个<shape>语句和<padstack\_def>语句组成。

6.4.2 语句组成

```
<global>语句包含以下内容：
    <unit>
    <shape>
    <padstack_def>
    <extensions>
```

6.4.3 unit 语句

6.4.3.1 概述

<unit>语句定义了 在 LPB 格式文件中使用的单元系统。

```
<unit>
    [<distance> element]
    [<angle> element]
    [<area> element]
    [<time> element]
    [<resistance> element]
    [<capacitance> element]
    [<resistivity> element]
    [<temperature> element]
    [<voltage> element]
    [<current> element]
    [<power> element]
    [<inductance> element]
    [<frequency> element]
    [<impedance> element]
    [<thermal_conductivity> element]
    [<specific_heat_capacity> element]
    [<density> element]
    [<thermal_diffusivity> element]
    [<coefficient_of_thermal_expansion> element]
    [<dynamic_viscosity> element]
```

[<extensions> element]...  
</unit>

这些值说明如何解释 LPB 格式文件中的数字。

6.4.3.2 语句组成

6.4.3.2.1 distance 语句

<distance>语句定义 LPB 格式文件中的长度单位。

<distance unit="length\_unit " />

length\_unit 应为下列值之一：

- pm 皮米
- nm 纳米
- μm 微米
- mm 毫米
- m 米
- inch 英寸
- mil 密耳

如果未指定<distance>语句，则设置 m 为默认值。

6.4.3.2.2 angle 语句

<angle>语句定义了一个角度单位。

<angle unit="unit\_of\_angle " />

unit\_of\_angle 应为下列值之一：

- degree
- radian

如果未指定<angle>语句，则设置 degree 为默认值。

6.4.3.2.3 area 语句

<Area>语句定义了一个面积单位。

<area unit="unit\_of\_area " />

unit\_of\_area 应为下列值之一：

- pm2 平方皮米
- nm2 平方纳米
- μm2 平方微米
- mm2 平方毫米
- m2 平方米
- i2 平方英寸

如果没有指定<area>语句，则设置 m2 为默认值。

6.4.3.2.4 time 语句

<time>语句定义一个以秒为单位的时间单位。

<time unit="unit\_of\_time " />

*unit\_of\_time* 应为下列值之一：

- ps 皮秒
- ns 纳秒
- us 微秒
- ms 毫秒
- s 秒

如果未指定<time>语句,则设置 s 为默认值。

6.4.3.2.5 resistance 语句

<resistance>语句定义了电阻率单位。

<resistance unit="unit\_of\_resistance " />

*unit\_of\_resistance* 应为下列值之一：

- fohm 飞欧
- pohm 皮欧
- nohm 纳欧
- uohm 微欧
- mohm 毫欧
- ohm 欧姆
- kohm 千欧
- Mohm 兆欧
- Gohm 吉欧

如果未指定<resistance>语句,则设置 ohm 为默认值。

6.4.3.2.6 capacitance 语句

<capacitance>语句定义了电容单位。

<capacitance unit="unit\_of\_capacitance " />

*unit\_of\_capacitance* 应为下列值之一：

- fF 飞法
- pF 皮法
- nF 纳法
- uF 微法
- mF 毫法
- F 法拉
- kF 千法
- mF 兆法
- gF 吉法

如果未指定<capacitance>语句,则设置 F 为默认值。

6.4.3.2.7 <resistivity>语句

<resistivity>语句定义了电阻率单位。

<resistivity unit="unit\_of\_resistivity " />

*unit\_of\_resistivity* 应为下列值之一：

- fohmm 飞欧米

nohmm	纳欧米
pohmm	皮欧米
uohmm	微欧米
mohmm	毫欧米
ohmm	欧姆米
Kohmm	千欧米
Mohmm	兆欧米
Gohmm	吉欧米

如果未指定<resistivity>语句,则设置 ohmm 为默认值。

6.4.3.2.8 temperature 语句

<temperature>语句定义了温度单位。

<temperature unit="unit\_of\_temperature " />

unit\_of\_temperature 应为下列值之一:

- C 摄氏度
- K 开尔文

如果未指定<temperature>语句,则设置 C 为默认值。

6.4.3.2.9 voltage 语句

<voltage>元件定义了电压单位。

<voltage unit="unit\_of\_voltage " />

unit\_of\_voltage 应为下列值之一:

- pV 皮伏
- nV 纳伏
- μV 微伏
- mV 毫伏
- V 伏特
- kV 千伏

如果未指定<voltage>语句,则设置 V 为默认值。

6.4.3.2.10 current 语句

<current>语句定义了电流单位。

<current unit="unit\_of\_current " />

unit\_of\_current 应为下列值之一:

- pA 皮安
- nA 纳安
- uA 微安
- mA 毫安
- A 安培
- kA 千安

如果未指定<current>语句,则设置 A 为默认值。

6.4.3.2.11 power 语句

<power>语句定义了功率单位。

<power unit="unit\_of\_power " />

unit\_of\_power 应为下列值之一：

- pW 皮瓦
- nW 纳瓦
- μW 微瓦
- mW 毫瓦
- W 瓦特
- kW 千瓦

如果未指定<power>语句，则设置 W 为默认值。

6.4.3.2.12 inductance 语句

<inductance>语句定义了电感单位。

<inductance unit="unit\_of\_inductance " />

unit\_of\_inductance 应为下列值之一：

- fH 飞亨
- pH 皮亨
- nH 纳亨
- μH 微亨
- mH 毫亨
- H 亨利
- kH 千亨

如果未指定<inductance>语句，则设置 H 为默认值。

6.4.3.2.13 frequency 语句

<frequency>语句定义了频率单位。

<frequency unit="unit\_of\_frequency " />

unit\_of\_frequency 应为下列值之一：

- μHz 微赫
- mHz 毫赫
- Hz 赫兹
- kHz 千赫
- MHz 兆赫
- GHz 吉赫

如果未指定<frequency>语句，则设置 Hz 为默认值。

6.4.3.2.14 impedance 语句

<impedance>语句定义了阻抗单位。

<impedance unit="unit\_of\_impedance " />

unit\_of\_impedance 应为下列值之一：

- fohm 飞欧
- pohm 皮欧
- nohm 纳欧
- μohm 微欧

mohm 毫欧  
ohm 欧姆  
kohm 千欧  
Mohm 兆欧

如果未指定<impedance>语句,则设置 ohm 为默认值。

6.4.3.2.15 thermal\_conductivity 语句

<thermal\_conductivity>语句定义了导热系数单位。

<thermal\_conductivity unit="unit\_of\_thermal\_conductivity"/>

unit\_of\_thermal\_conductivity 应是下列值之一:

W/(m \* K) 瓦特每米开尔文

W/(cm \* K) 瓦特每厘米开尔文

如果未指定<thermal\_conductivity>语句,则设置 W/(m \* K)为默认值。

6.4.3.2.16 specific\_heat\_capacity 语句

<specific\_heat\_capacity>语句定义了比热容单位。

<specific\_heat\_capacity unit="unit\_of\_specific\_heat\_capacity"/>

unit\_of\_specific\_heat\_capacity 应是下列值之一:

J/(g \* K) 焦耳每克开尔文

J/(kg \* K) 焦耳每千克开尔文

如果未指定<specific\_heat\_capacity>语句,则设置 J/(kg \* K)为默认值。

6.4.3.2.17 density 语句

<density>语句定义了密度单位。

<density unit="unit\_of\_density"/>

unit\_of\_density 应是下列值之一:

kg/m3 千克每立方米

g/m3 克每立方米

g/cm3 克每立方厘米

如果未指定<density>语句,则设置 kg/m3 为默认值。

6.4.3.2.18 thermal\_diffusivity 语句

<thermal\_diffusivity>语句定义了热扩散系数的单位。

<thermal\_diffusivity unit="unit\_of\_thermal\_diffusivity"/>

unit\_of\_thermal\_diffusivity 应是下列值之一:

m2/s 平方米每秒

cm2/s 平方厘米每秒

如果未指定<thermal\_diffusivity>语句,则设置 m2/s 为默认值。

6.4.3.2.19 coefficient\_of\_thermal\_expansion 语句

<coefficient\_of\_thermal\_expansion>语句定义了热膨胀系数的单位。

<coefficient\_of\_thermal\_expansion unit="unit\_of\_coefficient\_of\_thermal\_expansion"/>

unit\_of\_coefficient\_of\_thermal\_expansion 应是下列值之一:



1/K 每开尔文  
1/C 每摄氏度

如果未指定<coefficient\_of\_thermal\_expansion>语句,则设置 1/K 为默认值。

6.4.3.2.20 dynamic\_viscosity 语句

<dynamic\_viscosity>语句定义了动态黏度的单位。  
<dynamic\_viscosity unit="unit\_of\_dynamic\_viscosity"/>  
unit\_of\_dynamic\_viscosity 应是下列值之一:

kg/ms 千克每毫秒  
Pas 帕斯卡秒( $\text{Pas} = \frac{\text{千克}}{\text{毫秒}}$ )  
P 泊( $\text{P} = \frac{0.1 \text{ 千克}}{\text{毫秒}}$ )  
cP 厘泊( $\text{cP} = \frac{0.001 \text{ 千克}}{\text{毫秒}}$ )

如果未指定<dynamic\_viscosity>语句,则设置 kg/ms 为默认值。

6.4.3.3 例句

以下是<unit>语句的示例。

```
<unit>
  <distance unit="μm"/>
  <angle unit="degree"/>
  <area unit="μm2"/>
  <time unit="ps"/>
  <resistivity unit="ohmm"/>
  <temperature unit="°C"/>
  <voltage unit="V"/>
  <capacitance unit="pF"/>
  <resistance unit="mohm"/>
  <power unit="mW"/>
  <inductance unit="nH"/>
  <frequency unit="MHz"/>
  <impedance unit="ohm"/>
  <thermal_conductivity unit="W/(m * K)"/>
  <specific_heat_capacity unit="J/(kg * K)"/>
  <density unit="kg/m3"/>
</unit>
```

6.4.4 shape 语句

6.4.4.1 概述

<shape>语句定义其他属性和语句引用的基本形状,例如<padstack\_def>语句。

```
<shape>
    [<rectangle> element]...
    [<circle> element]...
    [<polygon> element]...
    [<extensions> element]...
</shape>
```

定义的形状的作用域仅限于声明该形状的文件。<shape>语句的内容由 0 或更多的<rectangle>、<circle>和<polygon>语句组成。

6.4.4.2 语句组成

```
<shape>语句包含以下内容：
<rectangle>
<circle>
<polygon>
<extensions>
```

6.4.4.3 rectangle 语句

6.4.4.3.1 概述

<rectangle>语句定义具有宽度、高度和旋转角的矩形的形状。

```
<rectangle
    id="identifier "
    width="width "
    height="height "
    [angle="rotation_angle "]
/>
```

定义的矩形的参考点位于矩形的中心，它用于确定矩形的旋转和位置。

6.4.4.3.2 定义属性

- <rectangle>语句的属性定义如下：
- a) id:此属性指定用于从其他属性和语句引用形状的唯一标识符；
  - b) width:此属性指定矩形的宽度，其单位由<unit>语句中的<distance>语句定义；
  - c) height:此属性指定矩形的高度，其单位由<unit>语句中的<distance>语句定义；
  - d) angle:此属性指定相对于矩形中心的逆时针旋转角度，如果没有指定角度，则将零设置为默认值，旋转角的单位由<unit>语句中的<angle>语句定义。

6.4.4.3.3 例句

矩形形状由以下代码表示，见图 12。

```
<rectangle id="bfrecl10" width="12" height="6" angle="30"/>
```

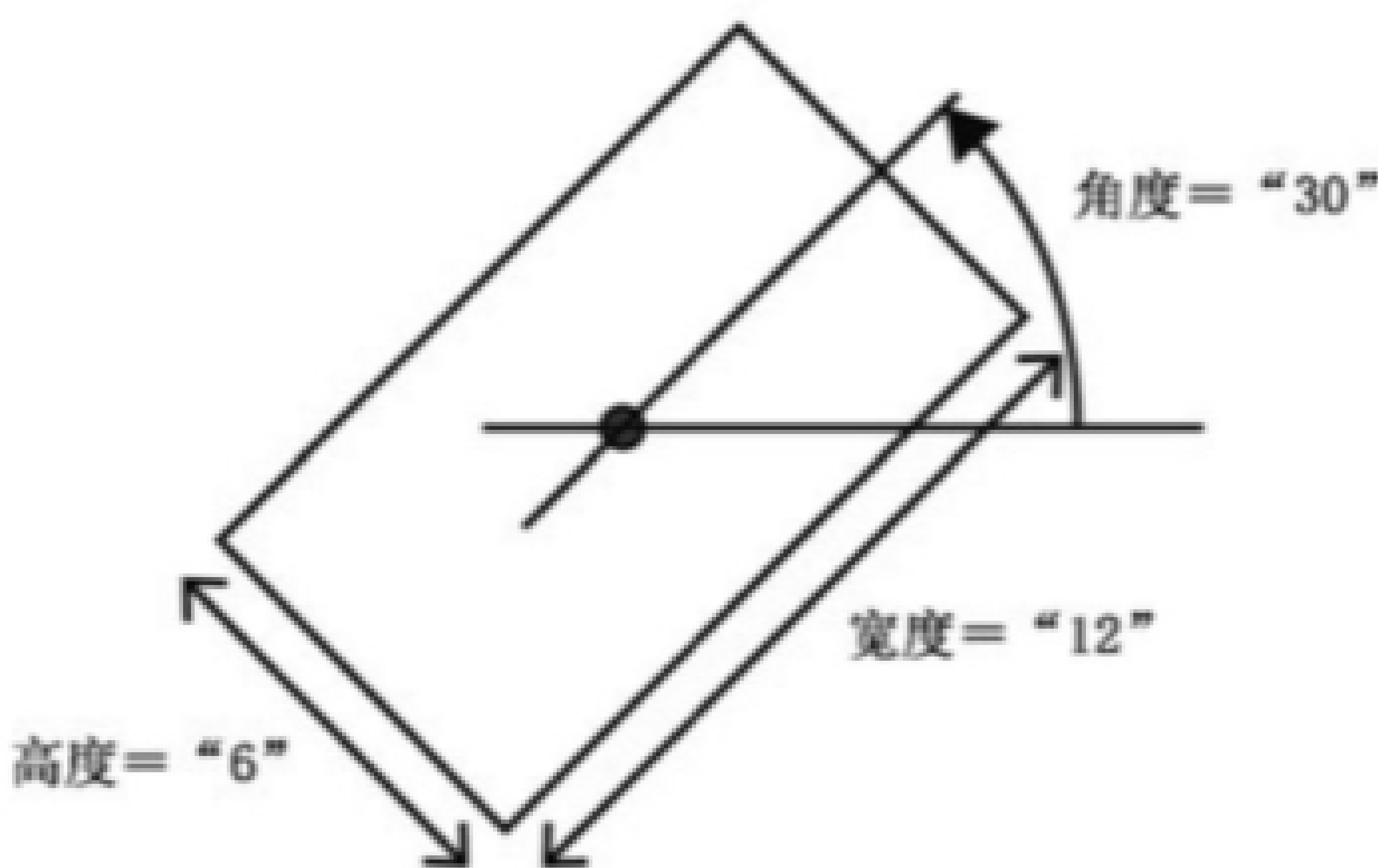


图 12 矩形形状示例

6.4.4.4 circle 语句

6.4.4.4.1 概述

〈circle〉语句按直径定义圆的形状。

```
〈circle
    id="identifier "
    diameter="diameter "
/〉
```

圆的参考点在中心。

6.4.4.4.2 定义属性

〈circle〉语句的属性定义如下：

- a) id:此属性指定用于从其他属性和语句引用形状的唯一标识符；
- b) diameter:此属性指定圆的直径,其单位由〈unit〉语句中的〈distance〉语句定义。

6.4.4.4.3 例句

圆形由以下代码表示,见图 13。

```
〈circle id="viapad12" diameter="12"/〉
```

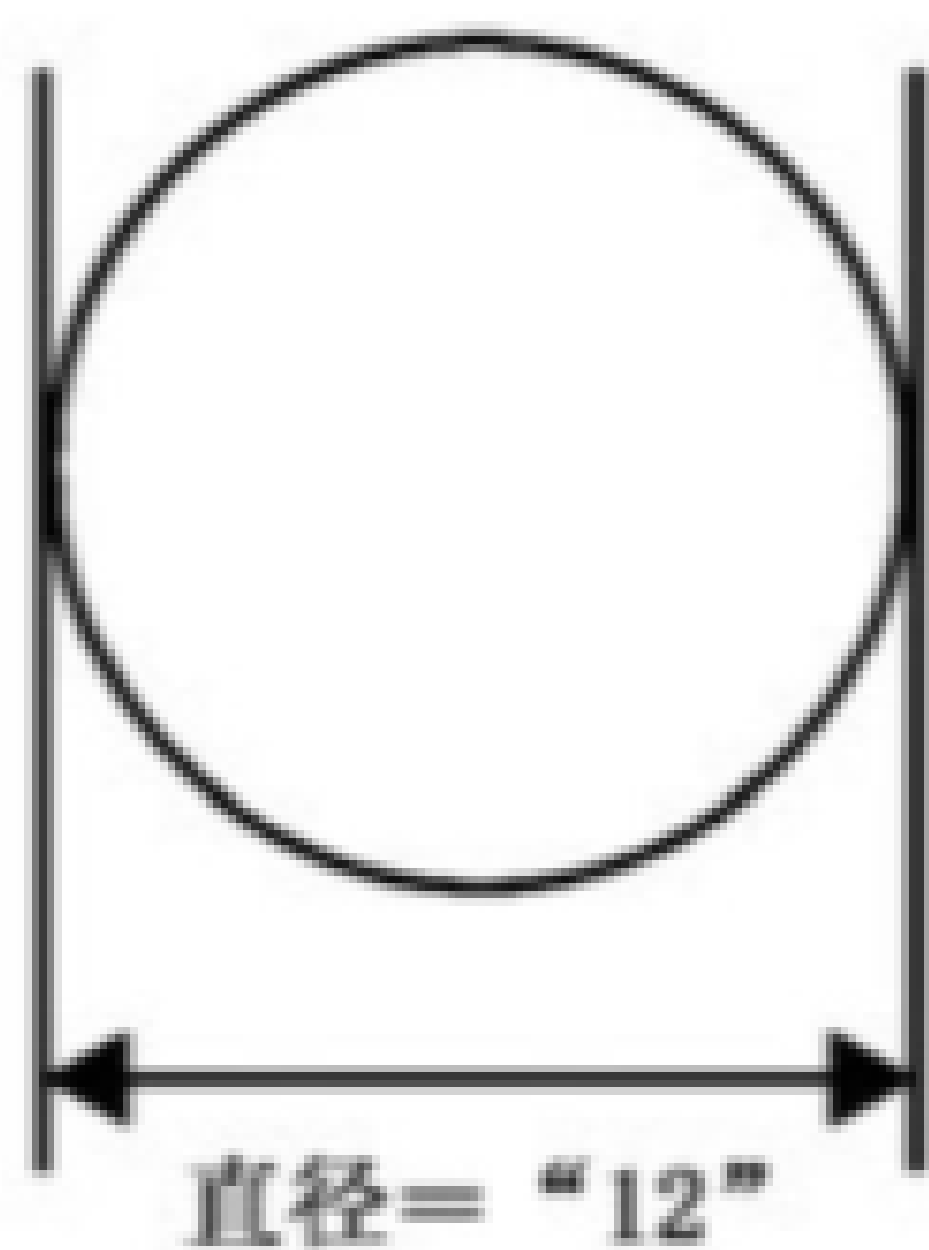


图 13 圆形示例

6.4.4.5 polygon 语句

6.4.4.5.1 概述

〈polygon〉语句定义封闭多边形的形状。

```
<polygon
    id="Identifier "
    points="x1, y1, x2, y2, x3, y3, x4, y4 ... "
    [angle="rotation_angle "]
/>
```

多边形的参考点设置为(0,0)。

6.4.4.5.2 定义属性

<polygon>语句的属性定义如下：

- a) id:此属性指定用于从其他属性和语句引用形状的唯一标识符；
- b) points:此属性指定至少四个点的序列以生成封闭多边形,最后一点和第一点应相同,点用逗号(,)分隔,每个点是由逗号(,)分隔的一对 XY 坐标,距离的单位由<unit>语句中的<distance>语句定义；
- c) angle:此属性指定逆时针旋转相对于多边形的参考点的角度,如果没有指定角度,则将零设置为默认值,旋转角的单位由<unit>语句中的<angle>语句定义。

6.4.4.5.3 例句

多边形形状由以下代码表示,见图 14。

```
<polygon id="X1" points="5,3,3,4,-3,2,-1,-3,4,-1,5,3"/>
```

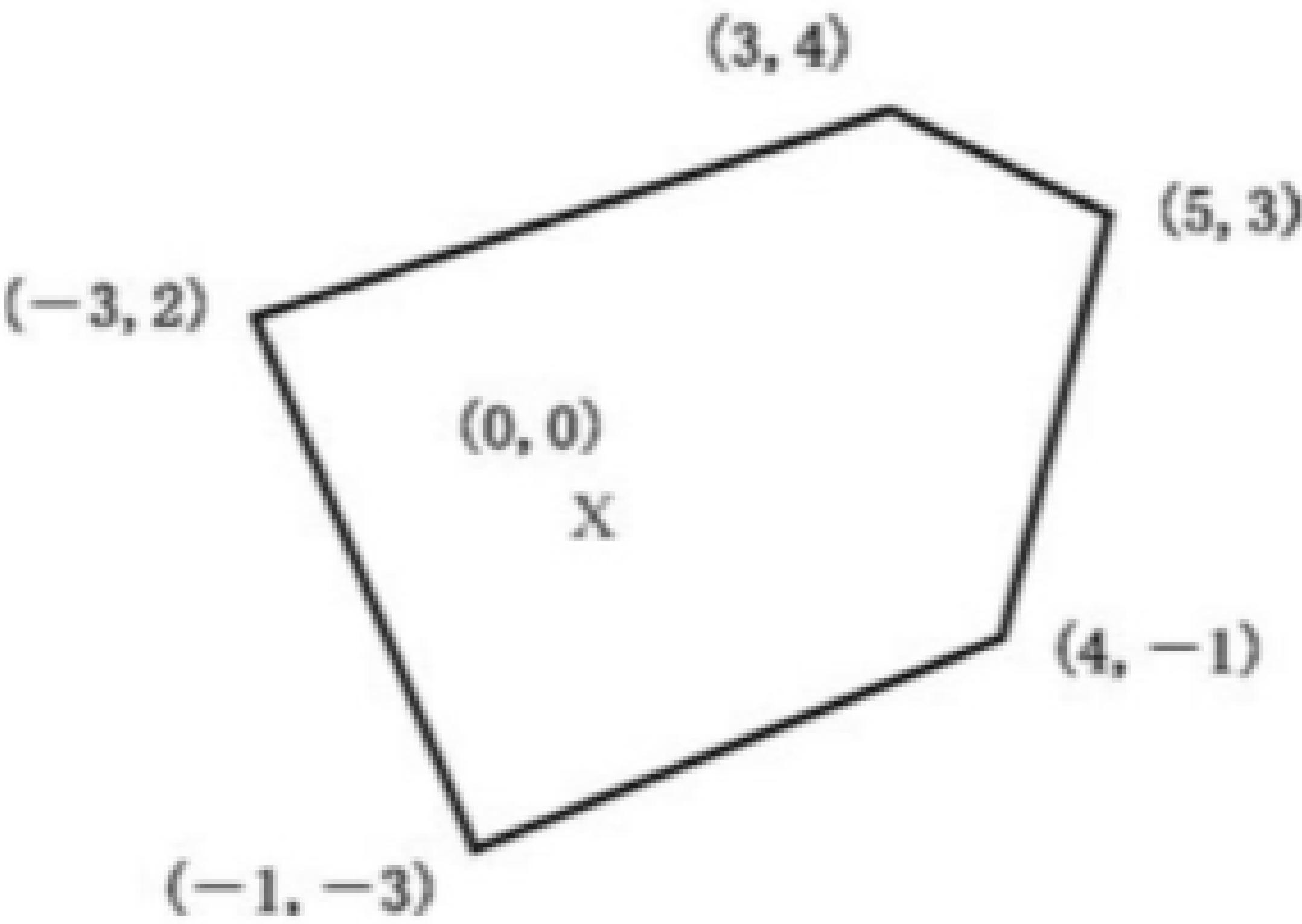


图 14 多边形形状示例

多边形形状旋转由以下代码表示,见图 15。

```
<polygon id="X2" points="5,3,3,4,-3,2,-1,3,4,-1,5,3" angle="30"/>
```

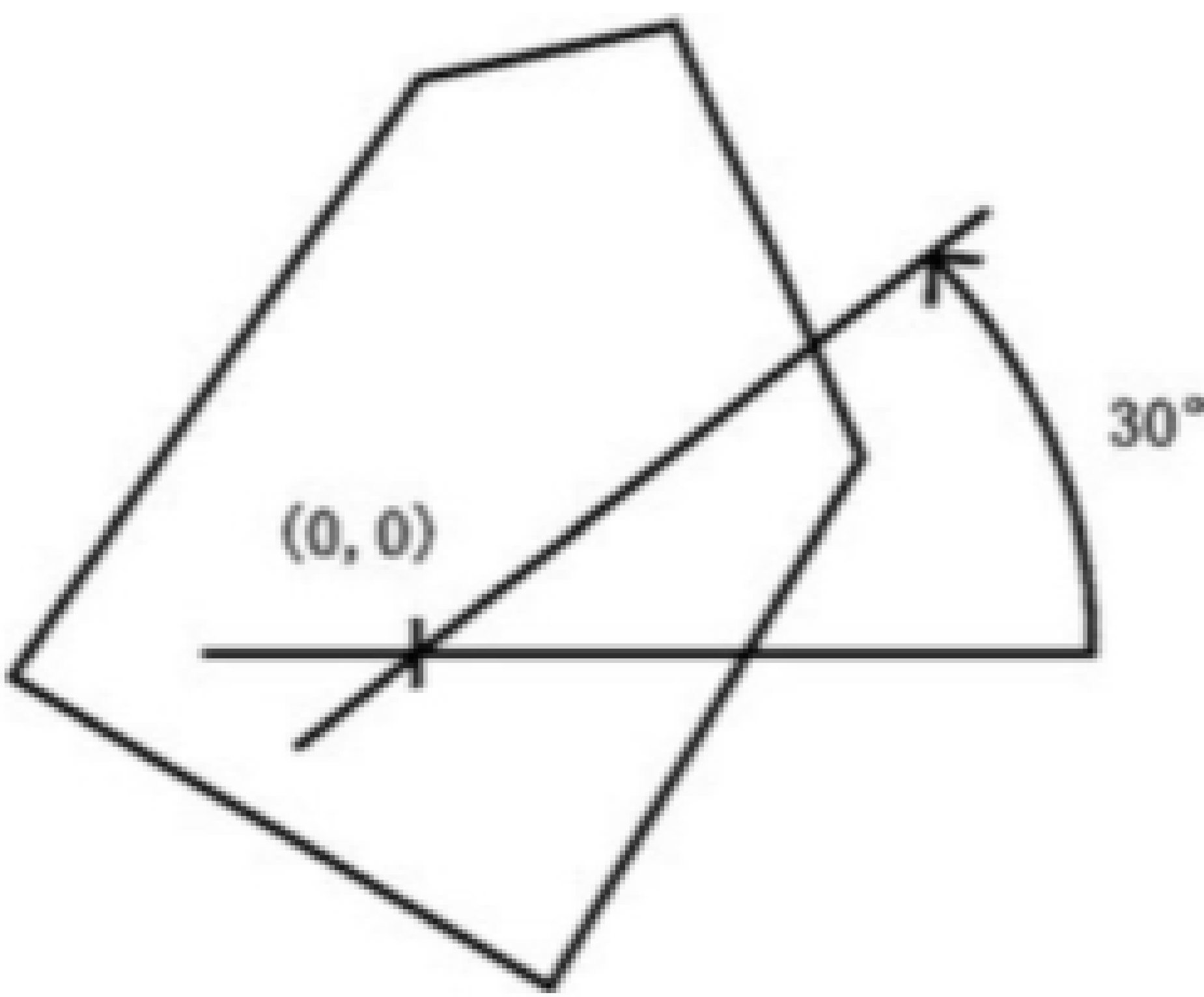


图 15 多边形形状的旋转示例

6.4.4.6 例句

以下是<shape>语句的示例。

```
<shape>
  <rectangle id="R1" width="100" height="10" />
  <circle id="C1" diameter="20" />
  <polygon id="P1" points="-10,-10,10,-10,10,10,-10,10,-10,-10" />
  <circle id="VLLPAD" diameter="500" />
  <circle id="VSLPAD" diameter="400" />
  <circle id="VLSPAD" diameter="700" />
  <circle id="VSAPAD" diameter="600" />
  <circle id="SHP.10" diameter="200" />
  <rectangle id="SHP.21" width="7500" height="10900" />
  <polygon id="SQRT" points="10,10,20,10,20,20,20,10,10,10" />
</shape>
```

6.4.5 padstack\_def 语句

6.4.5.1 概述

<padstack\_def>语句定义焊盘参考其他属性和语句。

```
<padstack_def >
  {<padstack> element}...
  {<extensions> element}...
</padstack_def>
```

<padstack\_det>语句由一个或多个<padstack>语句和零个或多个<extensions>语句组成。

6.4.5.2 语句组成

<padstack\_def>语句包含以下内容：

- <padstack>
- <extensions>

6.4.5.3 padstack 语句

6.4.5.3.1 概述

<padstack>语句定义了印制板(PWB)或封装的每个引脚或过孔的几何描述。

```
<padstack
  Id="identifier "
>
  {<ref_shape> element}...
  [<extensions> element]...
</padstack>
```

焊盘的参考点位于局部原点(0,0),焊盘定义的范围仅限于声明它的文件。〈padstack\_def〉语句的内容由一个或多个〈ref\_shape〉语句和零个或多个〈extensions〉语句组成。

6.4.5.3.2 定义属性

〈padstack〉语句的属性定义如下：  
id:此属性指定用于从其他属性和语句引用焊盘堆叠的唯一标识符。

6.4.5.4 ref\_shape 语句

6.4.5.4.1 概述

〈ref\_shape〉语句引用构造焊盘的形状。

```
〈ref_shape
    shape_id="identifier_of_referenced_shape "
    [type="shape_type "]
    [x="x_coordinate " y="y_coordinate "]
    [angel="rotation_angle "]
    [layer="shape_placed_layer "]
    [pad_layer="pad_placed_layer "]
/〉
```

引用的形状应在同一文件中的〈shape〉语句中定义。

6.4.5.4.2 定义属性

- 〈ref\_shape〉语句的属性定义如下。
- a) shape\_id:此属性指定在〈shape〉语句中定义的预定义形状的标识符,引用的形状应在同一文件中的〈shape〉语句中定义。
  - b) type:此属性指定形状如何用于导通孔结构,图 16 为一个导通孔结构示例,值应为下列之一:
    - 1) Antipad 指反焊盘或隔离,见图 6b);
    - 2) NonConnection 指无连接盘,见图 6b);
    - 3) Land 指正常焊盘,见图 6a)和图 6b);
    - 4) Drill 指钻孔和孔的外径,见图 6a)和图 6b);
    - 5) Hole 指成品孔和孔的内径,见图 6a)和图 6b)。
    - 6) SolderMask 指阻焊膜的形状;
    - 7) Resist 指阻焊剂的形状。

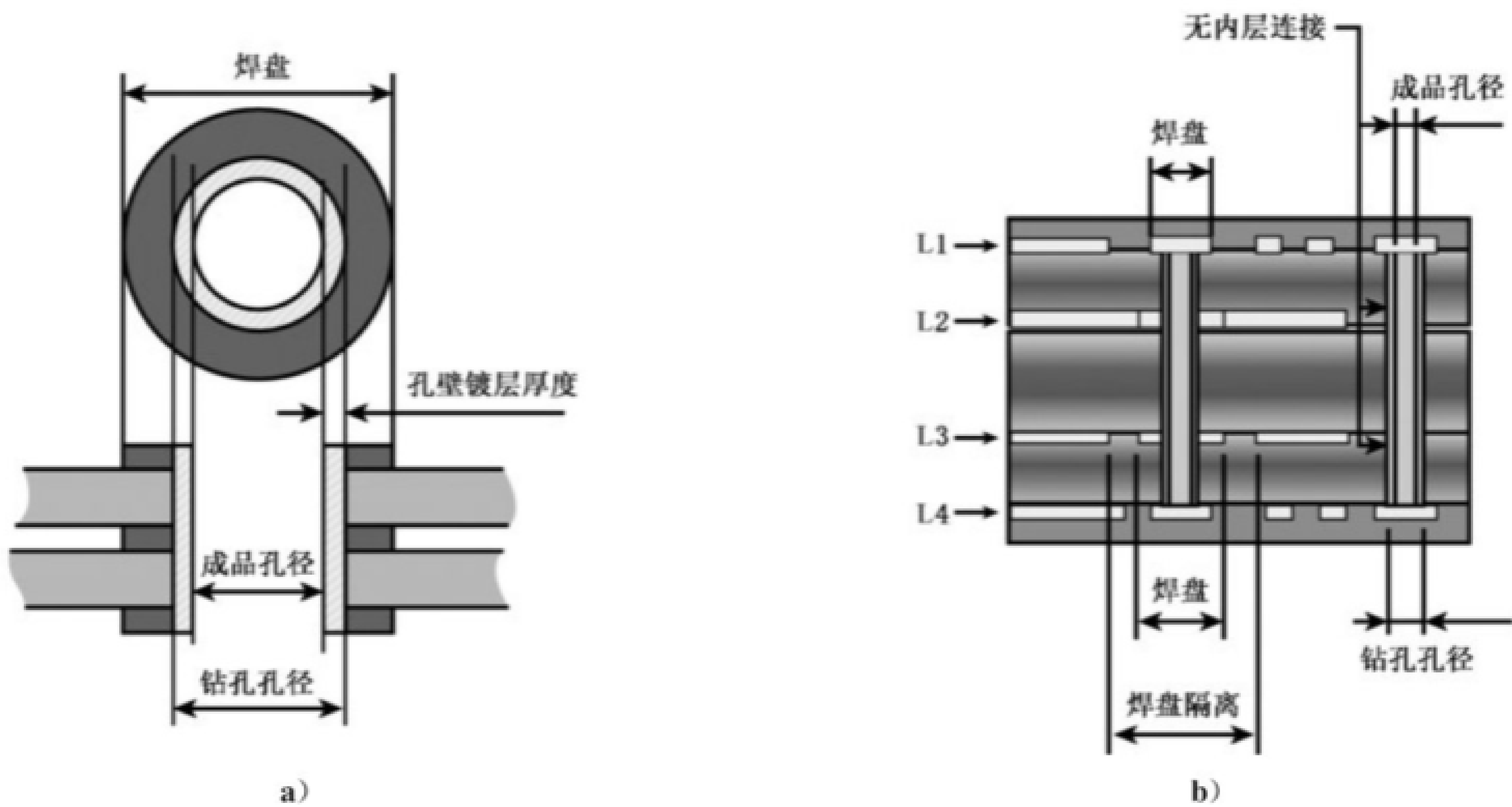


图 16 导通孔结构示例

- c) x,y:这些属性指定形状的参考点相对于原点的位置,x 和 y 属性分别指定 x 坐标和 y 坐标,如果未指定这些属性,则将零设置为默认值,坐标的单位由<unit>语句中的<distance>语句定义;
- d) angle:此属性指定逆时针旋转相对于形状的参考点的角度,如果没有指定角度,则将零设置为默认值,旋转角的单位由<unit>语句中的<angle>语句定义;
- e) layer:该属性指定在 R 格式中定义的图层的名称,该图形放置在该图层上;
- f) pad\_layer:此属性指定形状的放置侧,如图 17 所示,pad\_layer 属性被 C 格式独占使用,参数应为如下值之一:  
TOP 放置在顶部;  
BOTTOM 放置在底部;  
FOOTPRINT 焊盘引脚。

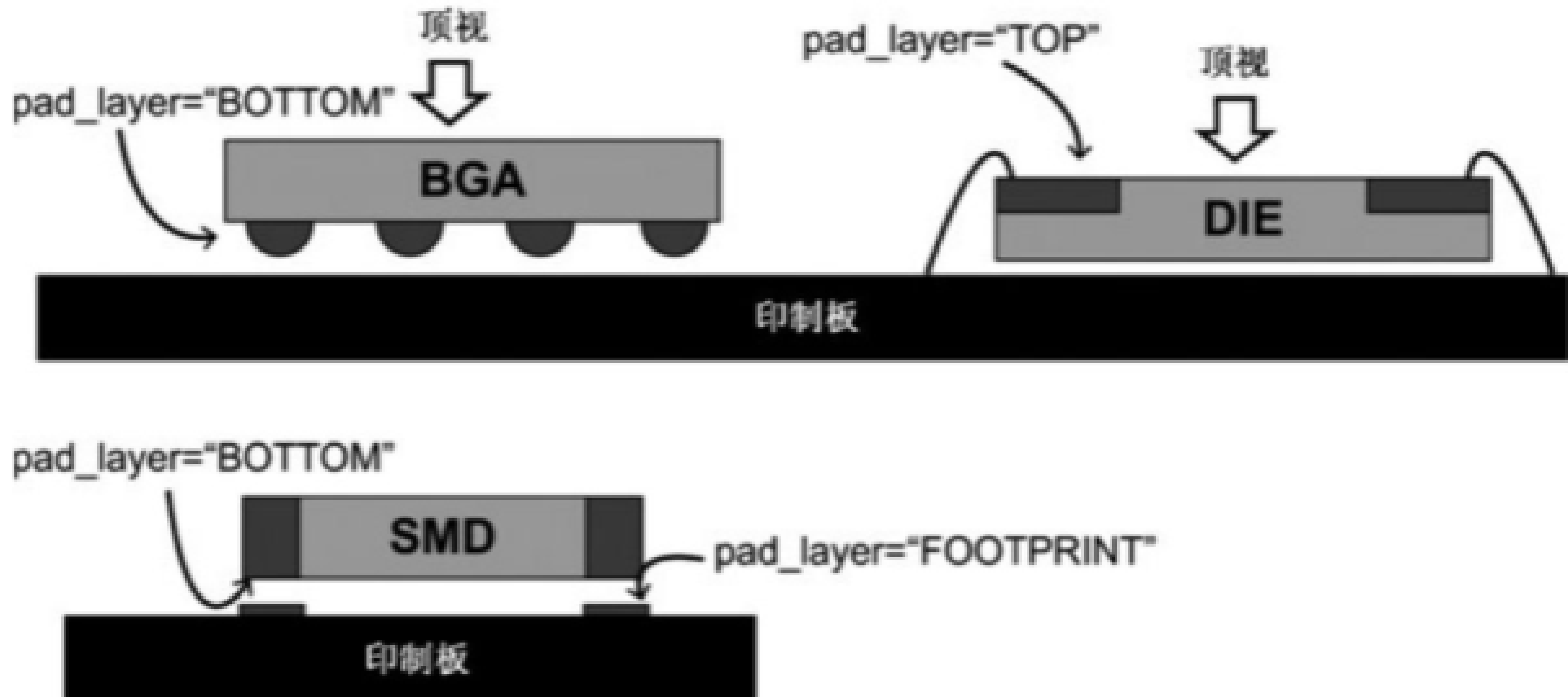


图 17 <pad\_layer>示意图

6.4.5.5 例句

以下是<padstack>语句的一个示例。

```
<padstack_def>  
  <padstack id="smallvia">
```

```

    <ref_shape shape_id="VLLPAD" />
    <ref_shape shape_id="VLAPAD"/>
    <ref_shape shape_id="VDRILL" />
    <ref_shape shape_id="VHOLE" />
    <ref_shape shape_id="VLLPAD" />
    <ref_shape shape_id="VLAPAD" />
    <ref_shape shape_id="VDRILL" />
    <ref_shape shape_id="VHOLE" />
  </padstack>
  <padstack id="ball" >
    <ref_shape shape_id="SHP.10"
              x="0" y="0" angle="0" pad_layer="BOTTOM"
    />
  </padstack>
</padstack_def>

```

7 M 格式

7.1 M 格式文件结构

M 格式文件的内容由一个<header>语句、零或多个<include>语句、零个或一个<current\_phase>语句和一个或多个<class>语句组成。各要素应按以下顺序加以规定：

```

<LPB_MFORMAT version="2020">
    <header> element
    [<include> element]...
    [<current_phase> element]
    {<class> element}...
    [<extensions> element]...
</LPB_MFORMAT>

```

7.2 include 语句

7.2.1 概述

<include>语句定义了要包含的其他 M 格式文件。

```

<include
    MFORMAT="name_of_M_format_file"
/>

```

7.2.2 定义属性

<include>语句的属性定义如下：

MFORMAT:此属性指定要包含的 M 格式文件的名称,当该属性不包含目录路径时,意味着 M 格式文件存在于当前目录中;该属性包含一个相对目录路径时,它意味着 M 格式文件存在于当前目录的相对位置;此外,该属性还可使用根目录中的绝对文件路径。



7.2.3 例句

以下是<include>语句的一个示例。

```
<include MFORMAT="LPBFMT-PKG.xml" />
<include MFORMAT="ydir/zdir/LPBFMT-PKG2.xml" />
<include MFORMAT="/TOPDIR/xdir/LPBFMT-SOC.xml" />
```

7.3 current\_phase 语句

7.3.1 通则

M 格式的一个目的是管理在几个设计阶段使用的 LPB 格式文件的版本。一个 M 格式便可包含这些 LPB 格式文件的版本。<current\_phase>语句用于声明当前设计阶段。处理 M 格式的应用程序在设计阶段应选择 LPB 格式文件。

```
<current_phase name="current_design_phase" />
```

7.3.2 定义属性

<current\_phase>语句的属性定义如下：  
Name:此属性指定由<class>语句中的 phase\_name 属性定义的设计阶段名称。

7.3.3 例句

下面的示例包括三个设计阶段。当前设计阶段为“step03\_cad”。应用程序应选择“step03\_cad”设计阶段使用的 LPB 格式文件。

```
<? xml version="1.0" ?>
<LPB_MFORMAT version="2020">
  <header project="myproj01"
    design_version="1.0"
    date="2018-06-17 13:00:00"
    author="Mr.X"
    company="JEITA/SDTC/LPBSC/2020TG"/>
  <current_phase name="step03_cad"/>
  <class name="myboard"
    phase_name="phase03_cad"
    comment="cad design ongoing"
    date="2018-06-14 13:00">
    ....
    ....
  </class>
  <class name="myboard"
    phase_name="phase02_sim"
    comment="si/pi simulation was ok"
    date="2018-06-07 13:00">
    ....
    ....
```

```
.....  
</class>  
<class name="myboard"  
  phase_name="phase01_fs"  
  comment="feasibility study done"  
  date="2018-06-01 13:00">  
.....  
.....  
</class>
```

7.4 class 语句

7.4.1 概述

<class>语句绑定形成功能组件(如 LSI 或印制电路板)的相关文件。

```
<class  
  [name="class_name "  
  [phase_name="design_phase_name "  
  [date="date "  
  [comment="comment_text "  
>  
  [<CFORMAT> element]  
  [<RFORMAT> element]...  
  [<GFORMAT> element]...  
  [<NFORMAT> element]...  
  [<OtherFile> element]...  
  [<ref_class> element]...  
  [<extensions> element]...  
</class>
```

7.4.2 定义属性

<class>语句的属性定义如下：

- a) name:此属性指定类的名称；
- b) phase\_name:此属性指定设计阶段名称,定义的 LPB 格式文件在<class>中指它们用于指定的设计阶段；
- c) date:此属性指定定义或修改类的日期；
- d) comment:此属性指定注释字符串。

7.4.3 语句组成

<class>语句包含以下内容：

```
<CFORMAT>  
<RFORMAT>
```

<GFORMAT>  
<NFORMAT>  
<OtherFile>  
<ref\_class>  
<extensions>

7.4.4 例句

以下是<class>语句的一个示例。

```
<class
  name="LPB_SAMPLE_CLASS"
  phase_name="Final_Design_Rev.1.0"
  date="20190331"
  comment="PWB "
>
  <CFORMAT
    comment="C-Format"
    file_name="CFMT_TOP.xml"
    design_revision="1.0"
    MD5="646da9ae5d90e6b51b0a578badae372f"
  />
  <RFORMAT
    comment="R-Format"
    file_name="RFMT_TOP.xml"
    design_revision="1.0"
    MD5="6ede01b9fed673b6619cffb5f96e1acf"
  />
  <GFORMAT
    comment="G-Format"
    file_name="GFMT_TOP.xml"
    MD5="6ede01b9fed672cfa746974210d68e96"
  />
  <NFORMAT
    comment="N-Format"
    file_name="NFMT_TOP.v"
    MD5="ee620bf842fb6646da9ae5d90e6b51b0"
  />
  <ref_class
    comment="Parts"
    file_name="./lib/LPB_PARTS.xml"
    class_name="parts_lib"
  />
</class>
```

7.4.5 CFORMAT 语句

7.4.5.1 概述

〈CFORMAT〉语句定义 LPB-元器件格式(C 格式)的文件。

```
〈CFORMAT
    [comment="comment_text "]
    file_name="name_of_C_format_file "
    design_revision="revision_number "
    [MD5="MD5_checksum "]
/>
```

7.4.5.2 定义属性

〈CFORMAT〉语句的属性定义如下：

- a) comment:此属性指定注释字符串；
- b) file\_name:此属性指定 C 格式文件的名称；
- c) design\_revision:此属性指定 C 格式文件的修订编号；
- d) MD5:此属性指定 C 格式文件的消息摘要算法 5(MD5)校验和。

7.4.5.3 例句

以下是〈CFORMAT〉语句的一个示例。

```
〈CFORMAT comment="C-Format" file_name="CFMT_TOP.xml"
    design_revision="1.0" MD5="5c34a4dd1bb48484e1e93eb5e23b3094" />
```

7.4.6 RFORMAT 语句

7.4.6.1 概述

〈RFORMAT〉语句定义 LPB-规则格式(R 格式)的文件。

```
〈RFORMAT
    [comment="comment_text "]
    file_name="name_of_R_format_file "
    design_revision="revision_number "
    [MD5="MD5_checksum "]x
/>
```

7.4.6.2 定义属性

〈RFORMAT〉语句的属性定义如下：

- a) comment:此属性指定注释字符串；
- b) file\_name:此属性指定 R 格式文件的名称；
- c) design\_revision:此属性指定 R 格式文件的修订编号；
- d) MD5:此属性指定 R 格式文件的 MD5 校验和。

7.4.6.3 例句

以下是<RFORMAT>语句的一个示例。

```
<RFORMAT comment="R-Format" file_name="RFMT_TOP.xml"
  design_revision="1.0" MD5="4d16098ad69f0a153387d6430a25806a" />
```

7.4.7 GFORMAT 语句

7.4.7.1 概述

<GFORMAT>语句定义 LPB-几何图形格式(G 格式)的文件。

```
<GFORMAT
  [comment="comment_text "]
  file_name="name_of_G_format_file "
  [MD5="MD5_checksum "]
/>
```

7.4.7.2 定义属性

<RFORMAT>语句的属性如下：

- a) comment:此属性指定注释字符串；
- b) file\_name:此属性指定 G 格式文件的名称；
- c) MD5:此属性指定 G 格式文件的 MD5 校验和。

7.4.7.3 例句

以下是<GFORMAT>语句的一个示例。

```
<GFORMAT comment="G-Format" file_name="GFMT_TOP.xfl"
  MD5="b3fed15159e9fbefdf67b603395eaf4c" />
```

7.4.8 NFORMAT 语句

7.4.8.1 概述

<NFORMAT>语句定义 LPB-网表(N 格式)的文件。

```
<NFORMAT
  [comment="comment_text "]
  file_name="name_of_N_format_file "
  [MD5="MD5_checksum "]
/>
```

7.4.8.2 定义属性

<NFORMAT>语句的属性定义如下：

- a) comment:此属性指定注释字符串；

- b) file\_name:此属性指定 N 格式文件的名称;
- c) MD5:此属性指定 N 格式文件的 MD5 校验和。

7.4.8.3 例句

以下是<NFORMAT>语句的一个示例。

```
<NFORMAT comment="N-Format" file_name="NFMT_TOP.v"
  MD5="343490687786f1420958e9aed2f2895b" />
```

7.4.9 OtherFile 语句

7.4.9.1 概述

<OtherFile>语句指定用户定义文件的名称。

```
<OtherFile
  [comment="comment_text "]
  file_name="name_of_format_file "
  [MD5="MD5_checksum "]
/>
```

7.4.9.2 定义属性

<OtherFile>语句的属性定义如下:

- a) comment:此属性指定注释字符串;
- b) file\_name:此属性指定用户定义文件的名称;
- c) MD5:此属性为用户定义的文件指定 MD5 校验和。

7.4.9.3 例句

以下是<OtherFile>语句的示例。

```
<OtherFile comment="Power Model" file_name="DDRPowerModel.sp"
  MD5="7ca273b0993527d8df5deed246b8fbff" />
```

7.4.10 ref\_class 语句

7.4.10.1 概述

<ref\_class>语句用于引用相关的其他<class>语句,例如与电路板设计相对应的元器件组装。它指定了由<class>语句中的 name 属性定义的类名。

```
<ref_class
  [comment="comment_text"]
  [file_name="name_of_MFormat_file"]
  class_name="reference_class_name"
>
  [<extensions>element]...
</ref_class>
```

7.4.10.2 属性定义

- 〈ref\_class〉语句的属性定义如下：
- a) comment:此属性指定注释字符串；
  - b) file\_name:此属性指定要引用的 M 格式文件的名称,指定此属性后,搜索相关类的范围将扩展到指定的 M-格式文件；
  - c) class\_name:此属性指定类名以引用相关的其他类,例如与板设计相对应的元器件组。

8 C 格式

8.1 C 格式文件结构

C 格式文件的内容由一个〈header〉和〈global〉语句、一个或多个〈model〉语句、一个或零〈Component〉语句和零个或多个〈extensions〉语句组成。应按以下顺序规定这些要素：

```
〈LPB_CFORMAT version="2020"〉
    〈header〉 element
    〈global〉 element
    {〈module〉 element}...
    [〈component〉 element]
    [〈extensions〉 element]...
〈/LPB_CFORMAT〉
```

8.2 module 语句

8.2.1 概述

组件是半导体芯片或 LSI 封装的基本设计单元。〈module〉语句包含了组件的几何信息、设计约束、I/O 接口和电气模型。

```
〈module
    name="name_of_module "
    type="module_type "
    [shape_id="identifier_of_referenced_shape "]
    [x="x_coordinate " y="y_coordinate "]
    [angle="angle "]
    [thickness="thickness "]
    [ref_rule_name="referenced_design_rule_name "]
〉
    [〈size_code〉 element]
    [〈socket〉 element]...
    [〈guideline〉 element]
    [〈specification〉 element]
    [〈surrounding_characteristics〉 element]
    [〈blockage〉 element]
    [〈component〉 element]
```

```
[<keepaway> element]
{<reference> element}...
[<extensions> element]...
</module>
```

<module>语句的内容由零或多个<socket>、<reference>和<extensions>语句以及一个或零<guide-line>、<specification>、<surrounding\_characteristics>、<blockage>、<keepaway>和<component>语句组成。

<socket>语句定义 I/O 端口和设计约束。

<specification>语句定义了组件本身的规范,例如功耗。<surrounding\_characteristics>语句定义了三维结构的物理规格,例如热导率。

<reference>语句将<socket>语句中定义的端口与在其他文件中定义的电节点或物理数据关联起来。例如,<reference>语句能为没有物理信息的 IBIS 的 I/O 节点提供位置。

8.2.2 定义属性

<module>语句的属性如下。

- a) name:此属性定义用于从其他属性和语句引用组件的组件名称,组件名称在相同的 C 格式文件中是唯一的。
- b) type:此属性定义组件类型,参数应为下列之一:  
LSI:半导体集成电路;  
PKG:封装;  
PWB:印制线路板或印制电路板;  
C:电容器;  
R:电阻器;  
L:电感器;  
OTHER:其他类型。  
如果未指定,设置 OTHER 为默认值。
- c) shape\_id:此属性定义预定义形状的标识符,以定义组件的边界形状,指定的形状应在同一文件中的<shape>语句处定义。
- d) x/y:此属性定义形状相对于原点的参考点位置。x 和 y 属性分别指定 x 坐标和 y 坐标,组件放在指定的点上(参见图 18),如果未定义这些属性,则将零设置为默认值,坐标的单位由<unit>语句中的<distance>语句定义。

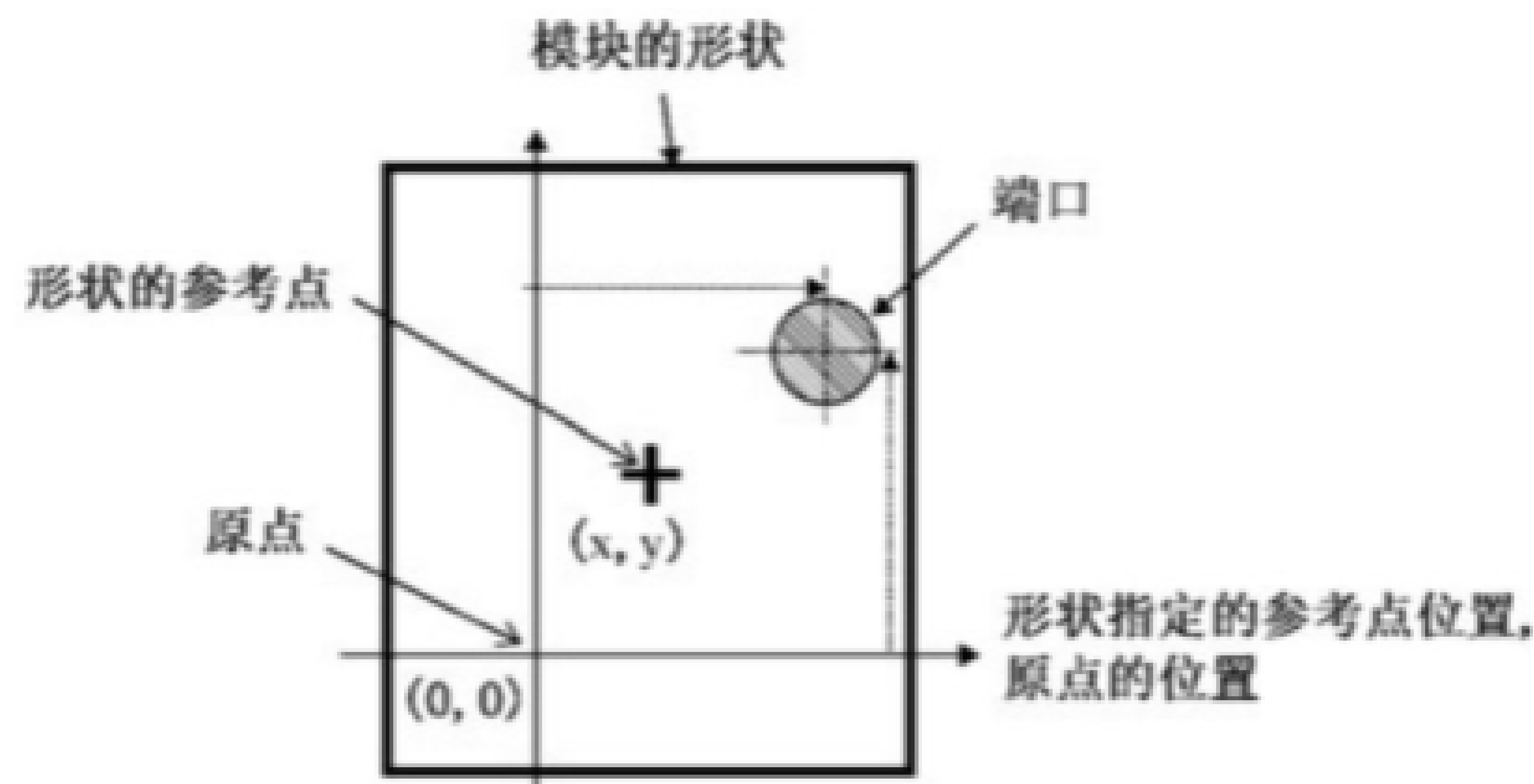


图 18 组件的形状和布局示例



- e) `angle`:此属性定义逆时针旋转相对于形状的参考点的角度,如图 19 所示。如果没有定义角度,则将零设置为默认值,旋转角的单位由`<unit>`语句中的`<angle>`语句定义。

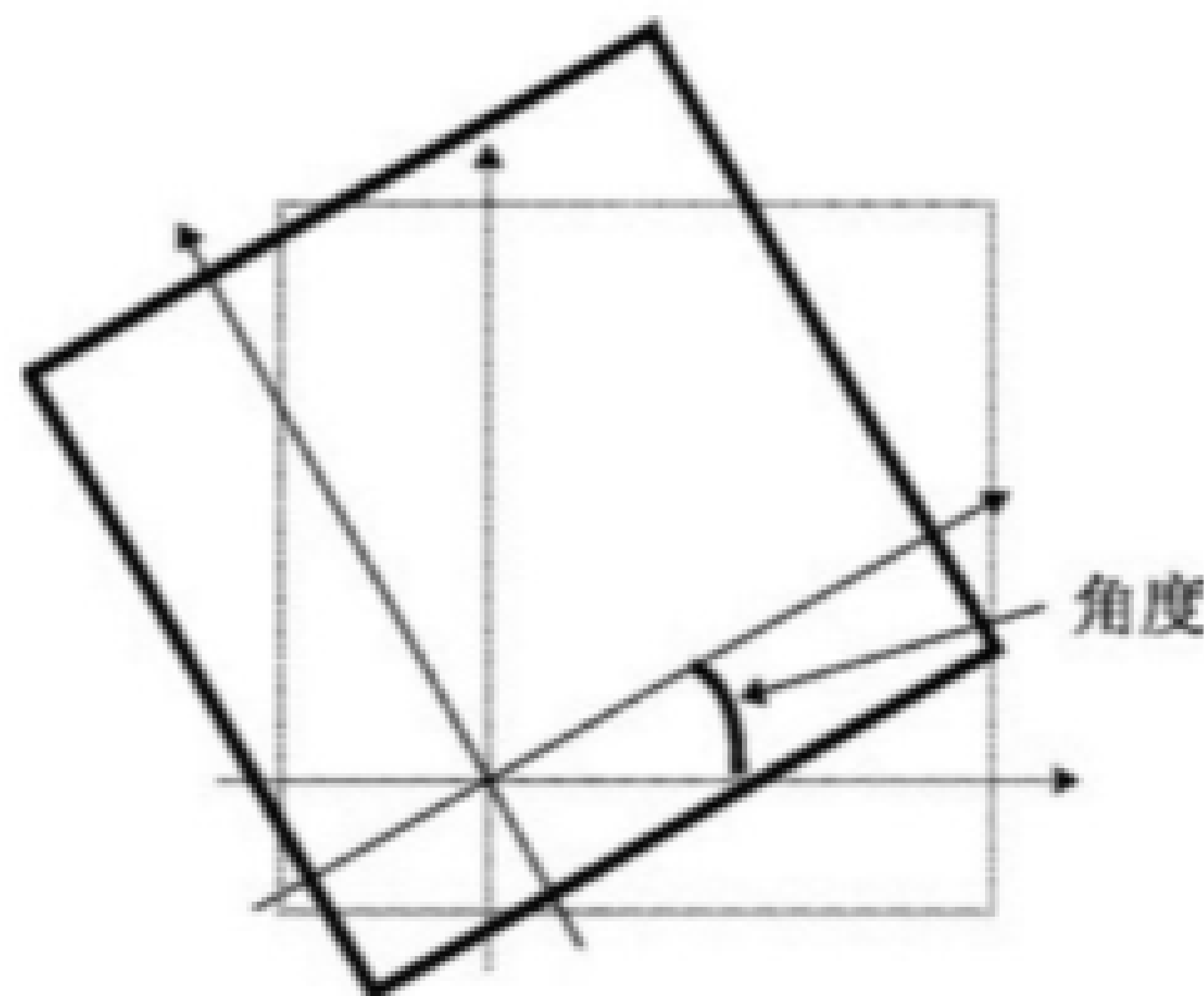


图 19 组件旋转示意图

- f) `thickness`:此属性定义不包括引脚或焊球的组件厚度,如图 20 所示。如果未定义厚度,则将零设置为默认设置,坐标的单位由`<unit>`语句中的`<distance>`语句定义。

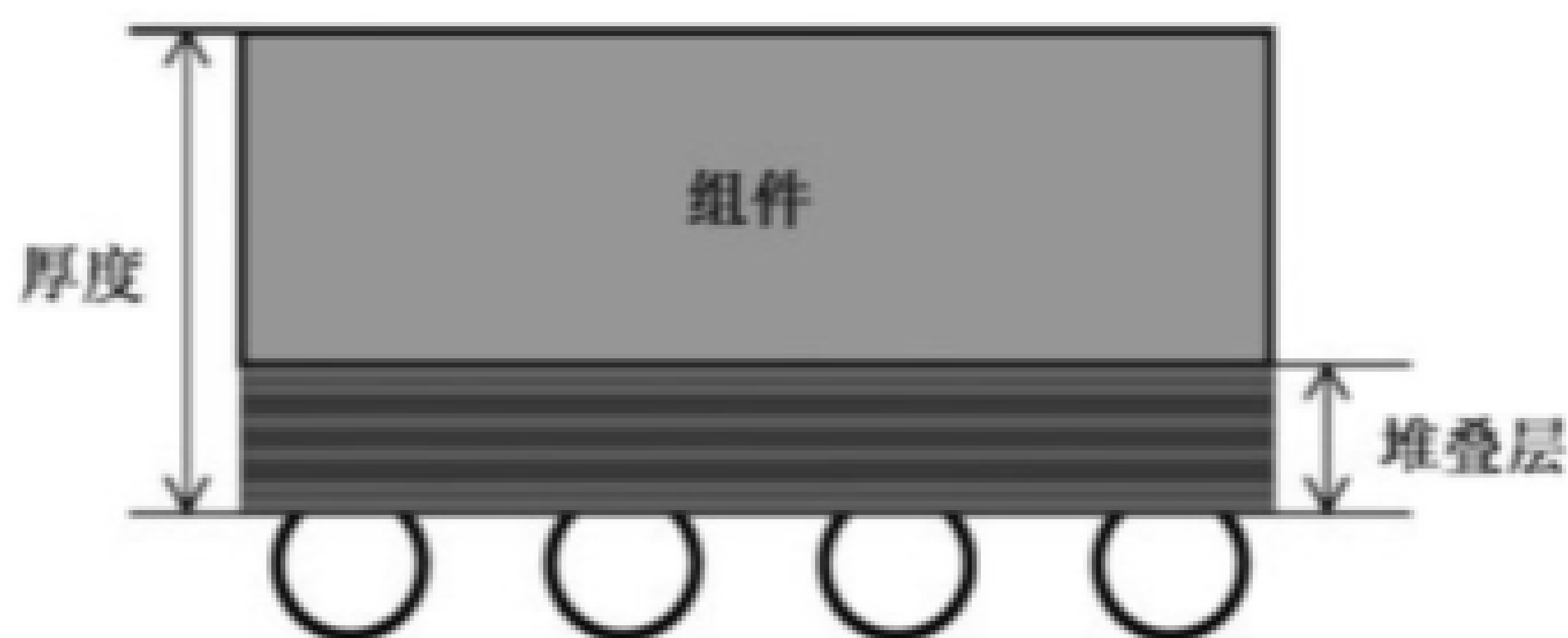


图 20 组件厚度属性示意图

- g) `ref_rule_name`: 此属性指定要应用于此模块的设计规则的名称,该名称由 R 格式的`<physicaldesign>`语句定义。

8.2.3 例句

以下是`<module>`语句的一个示例。

```
<global>
  <shape>
    <rectangle id="PFBGABODY" width="1200" height="1200" />
    <circle id="B500" circle="500" />
  </shape>
  <padstack_def>
    <padstack id="BGABALL" type="BALLPAD">
      <ref_shape shape_id="B500" x="0" y="0" />
    </padstack>
  </padstack_def>
</global>
<module name="BGA" type="PKG">
  shape_id="PFBGABODY" x="0" y="0" thickness="540">
    <socketname="BGAIO">
      <default>
        <port_shape padstack_id="BGABALL" />
      </default>
```

```
        <port id="A1" x="-1100" y="-1100" />
        <port id="A2" x="-1000" y="-1100" />
    </socket>
</module>
```

8.2.4 size\_code 语句

8.2.4.1 概述

<size\_code>语句定义表面安装器件(SMD)的大小。SMD 的大小由一个代码表示,例如 0603。该代码包含封装的宽度和高度,以英制或公制单位表示。

```
<size_code
    [metric="metric_size_code"]
    [imperial="imperial_size_code"]
/>
```

8.2.4.2 属性定义

- <size\_code>语句的属性定义如下:
- a) Metric:此属性指定了以公制单位给出的尺寸代码;
  - b) Imperial:此属性指定了以英制单位给出的尺寸代码。

8.2.4.3 例句

以下是<size\_code>语句的示例。

```
<module name="MAG2096xp" type="C"
    shape_id="smd0603" x="0" y="0" thickness="0.45">
    <size_code imperial="0201" metric="0603"/>
    .. .. .
</module>
```

8.2.5 socket 语句

8.2.5.1 概述

<socket>语句定义 I/O 端口和设计约束。

```
<socket
    name="socket_name "
>
    [ <default> element ]
    { <port> element }...
    [ <portgroup> element ]...
    [ <powerdomain_group> element ]...
    [ <swappable_port> element ]...
    [ <swappable_group> element ]...
    [ <port_assignment > element ]...
    [ <frequency> element ]...
    [ <constraint> element ]...
```

```
[ <extensions> element ]...  
</socket>
```

端口定义既是逻辑信息,又是几何信息,逻辑信息包括信号方向、名称和类型,几何信息包括端口的形状和位置,就像脚印一样,设计约束为连接组件的信号线提供了约束条件。

<socket>语句包含以下内容:

```
<default>  
<port>  
<portgroup>  
<powerdomain_group>  
<swappable_port>  
<swappable_group>  
<frequency>  
<constraint>  
<extensions>
```

<port>语句定义 I/O 端口,<default>语句定义端口的默认形状。每个端口可有不同的形状,但是如果所有端口都具有相同的形状,则可使用<default>语句。

<portgroup>语句定义由同一文件中的其他语句引用的端口集。

<powerdomain\_group>语句定义了从组件端口进出的信号的电源域,<swappable\_port>和<swappable\_group>语句相互定义可交换端口集,例如双倍数据速率 3 同步动态随机存取存储器(DDR 3 SDRAM)数据总线字节,<frequency>语句定义从组件端口进出的信号的工作频率,<constraint>语句定义设计约束,例如倾斜的限制。

图 21 显示了底座的示例。在 BGA 封装的情况下,底座是一组焊锡球,一个<module>语句可能有多个<socket>语句。图 21b)是含印制板的示例,在本例中,单卡边缘和两个连接器被定义为底座。

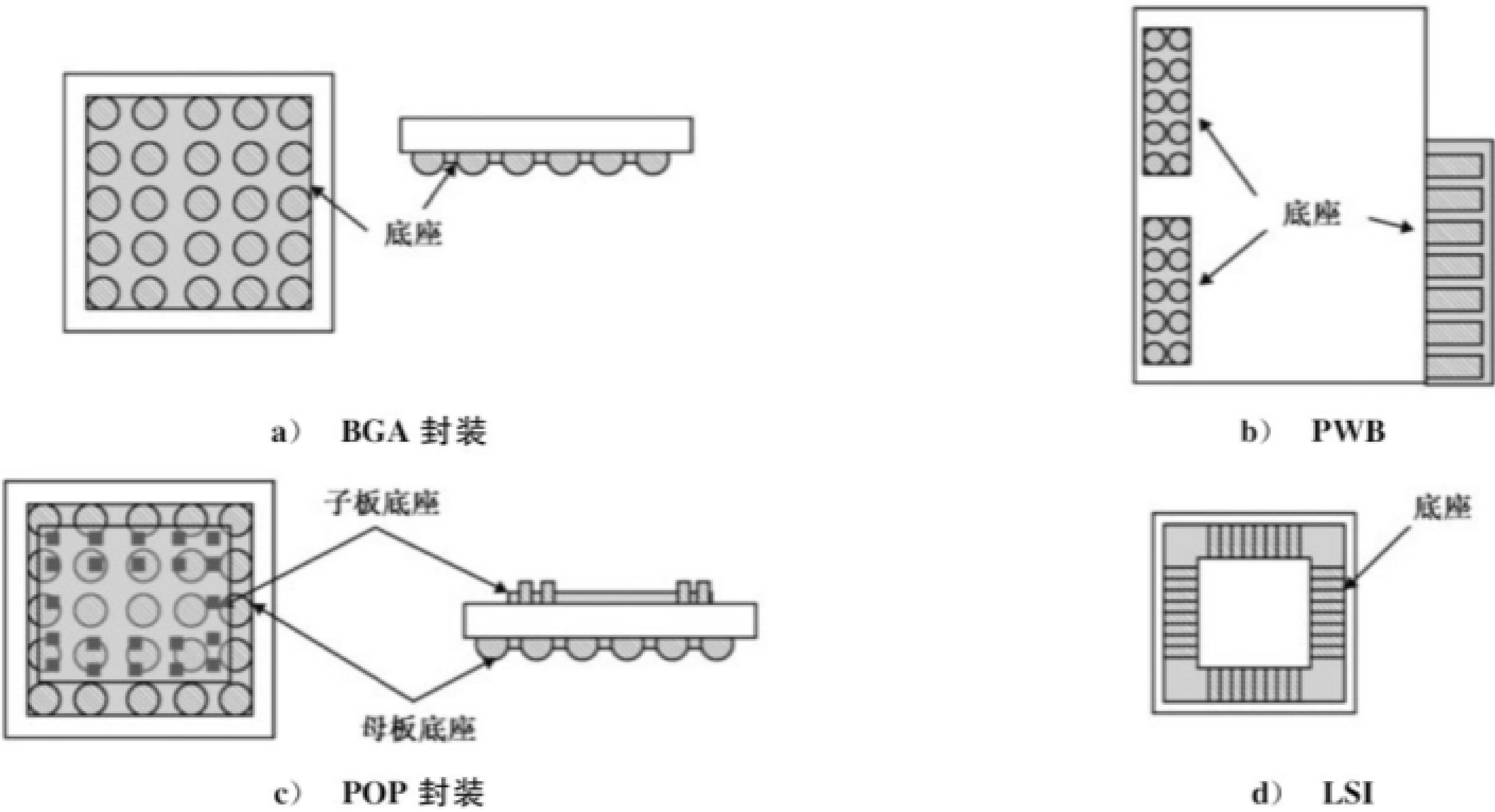


图 21 底座和组件结构示例

8.2.5.2 定义属性

〈socket〉语句的属性定义如下：

name：此属性定义用于从其他属性和语句引用底座的底座名称，底座名称在相同的 C 格式文件中是唯一的。

8.2.5.3 语句组成

〈socket〉语句包含以下内容：

- 〈default〉
- 〈port〉
- 〈portgroup〉
- 〈powerdomain\_group〉
- 〈swappable\_port〉
- 〈swappable\_group〉
- 〈port\_assignment〉
- 〈frequency〉
- 〈constraint〉
- 〈extensions〉

8.2.5.4 default 语句

8.2.5.4.1 概述

〈default〉语句定义在〈port〉语句中未定义形状时使用的端口的默认形状。

```
〈default〉  
    [〈port_shape〉 element]  
    [〈ball_shape〉 element]  
    [〈extensions〉 element]...  
〈/default〉
```

〈default〉语句的内容由 0 个或 1 个〈port\_form〉和〈ball\_form〉语句组成，它们分别定义了焊盘形状和焊球形状。

8.2.5.4.2 port\_shape 语句

8.2.5.4.2.1 概述

〈port shape〉语句定义 I/O 端口的默认焊盘形状。

```
〈port_shape padstack_id="identifier_of_referenced_padstack" /〉
```

在〈port〉语句中未定义焊盘形状时使用默认值。

## 8.2.5.4.2.2 定义属性

〈port\_shape〉语句的属性定义如下：

padstack\_id:此属性指定用于定义默认的预定义焊盘形状的标识符,在同一个文件中的〈padstack\_def〉语句中,将定义引用的焊盘。

## 8.2.5.4.2.3 例句

以下是〈port\_shape〉语句的一个示例。

```

〈global〉
  〈shape〉
    〈circle id="circ_3" diameter="1500" /〉
    〈circle id="circ_4" diameter="750" /〉
  〈/shape〉
  〈padstack_def〉
  〈padstack id="PAD.4"〉
    〈ref_shape shape_id="circ_3" type="Land"
      x="0" y="0" pad_layer="TOP" /〉
    〈ref_shape shape_id="circ_4" type="Hole"
      x="0" y="0" pad_layer="TOP" /〉
    〈ref_shape shape_id="circ_3" type="Land"
      x="0" y="0" pad_layer="BOTTOM" /〉
    〈ref_shape shape_id="circ_4" type="Hole"
      x="0" y="0" pad_layer="BOTTOM" /〉
  〈/padstack〉
  〈/padstack_def〉
〈/global〉
〈module name="LPB_2012_SAMPLE" type="PWB" shape_id="SHAPE.1"
  x="0" y="0" angle="0"〉
  〈socket name="SMA_X1"〉
    〈default〉
      〈port_shape padstack_id="PAD.4" /〉
    〈/default〉
    〈port id="1" x="1570.7" y="32293.2" /〉
    〈port id="2" x="-429.3" y="34293.2" /〉
    〈port id="3" x="-429.3" y="30293.2" /〉
    〈port id="4" x="3570.7" y="30293.2" /〉
    〈port id="5" x="3570.7" y="34293.2" /〉
  〈/socket〉
〈/module〉

```

8.2.5.4.3 ball shape 语句

8.2.5.4.3.1 概述

⟨ball\_shape⟩语句定义 BGA 封装的 I/O 端口的默认焊球形状。

```
⟨ball_shape ball_name="name_of_referenced_ball" />
```

当焊球形状未在⟨port⟩语句中定义时,则使用默认值。

8.2.5.4.3.2 定义属性

⟨ball\_shape⟩语句的属性定义如下:

ball\_name:此属性指定用于定义焊球的默认形状的焊球名称,焊球特征参数在 R 格式文件中的⟨ball\_def⟩语句中定义。

8.2.5.4.3.3 例句

以下是⟨ball\_shape⟩语句的一个示例。

```
⟨ball_shape ball_name="BGA_BALL" />
```

8.2.5.5 port 语句

8.2.5.5.1 概述

⟨port⟩语句定义组件端口的逻辑和几何信息。

```
⟨port
    [id="identifier "
        [padstack_id="identifier_of_referenced_padstack "]
        [ball_name="name_of_reference_ball "]
        [x="x_coordinate" y="y_coordinate "]
        [angle="angle "]
    ]
    [name="port_name "
        [direction="signal_direction "]
        [type="signal_type "]
    ]
>
    [⟨impedance⟩ element]
    [⟨delay⟩ element]
    [⟨voltage⟩ element]
    [⟨max_current⟩ element]
    [⟨dumping⟩ element]
    [⟨decap⟩ element]
    [⟨extensions⟩ element]...
</port>
```

逻辑信息是端口名称、信号方向和信号类型,这些属性分别由名称、方向和类型属性定义;几何信息是端口标识符、端口形状和位置,这些属性由 padstack\_id、ball\_name、x、y、angle 属性定义。

⟨port⟩语句可以同时具有逻辑和几何信息,也可以只有一种类型的信息。⟨port⟩语句的内容由零个或一个⟨impedance⟩、⟨delay⟩、⟨voltage⟩、⟨max\_current⟩、⟨dumping⟩、⟨decap⟩和⟨extensions⟩语句组成,实际阻抗和延迟是由这些语句定义的。⟨voltage⟩和⟨max|\_Current⟩语句用于定义电压调节器或逻辑 IC 的输出电压和电流。⟨dumping⟩和⟨decap⟩语句用于定义如何放置转储电阻和解耦电容器。

8.2.5.5.2 定义属性

⟨port⟩语句的属性定义如下。

- a) id:此属性指定用于从其他属性和语句引用端口的唯一标识符,标识符在⟨socket⟩语句中是唯一的。通常,对于 BGA 封装,id 可能是 A1、A2、A3 等。
- b) padstack\_id:此属性指定用于定义端口预定义焊盘形状的标识符,在同一个文件中的⟨padstack\_def⟩语句中,如果未指定 padstack\_id,则使用⟨Default⟩语句中定义的默认形状。
- c) ball\_name:此属性用于 BGA 封装,用定义焊球的名称来指定焊球的形状,焊球特征参数在 R 格式文件中的⟨ball\_def⟩语句中定义。如果未指定 ball\_name,则使用相同⟨socket⟩语句中的⟨default⟩语句中定义的默认形状。
- d) x/y:这些属性指定引用的焊盘的参考点相对于组件的本地原点的位置,x 和 y 属性分别指定 x 坐标和 y 坐标,坐标的单位由⟨unit⟩语句中的⟨distance⟩语句定义。
- e) angle:此属性指定逆时针旋转相对于组件的局部原点的角度。如果没有指定角度,则将零设置为默认值,旋转角的单位由⟨unit⟩语句中的⟨angle⟩语句定义。
- f) name:此属性指定端口的名称。通常,端口名称与从端口输入/输出的信号相同,相同的名称可以用于不同的端口,例如,连接到同一地面平面的端口可以具有相同的端口名称。
- g) direction:此属性指定端口的信号方向,如图 22 所示。参数应为下列之一:  
in:接收进入组件的信号;端口;  
out:驱动信号输出组件的端口;  
inout:可以接受信号进出组件的端口;端口的电源和接地类型应是输入输出。

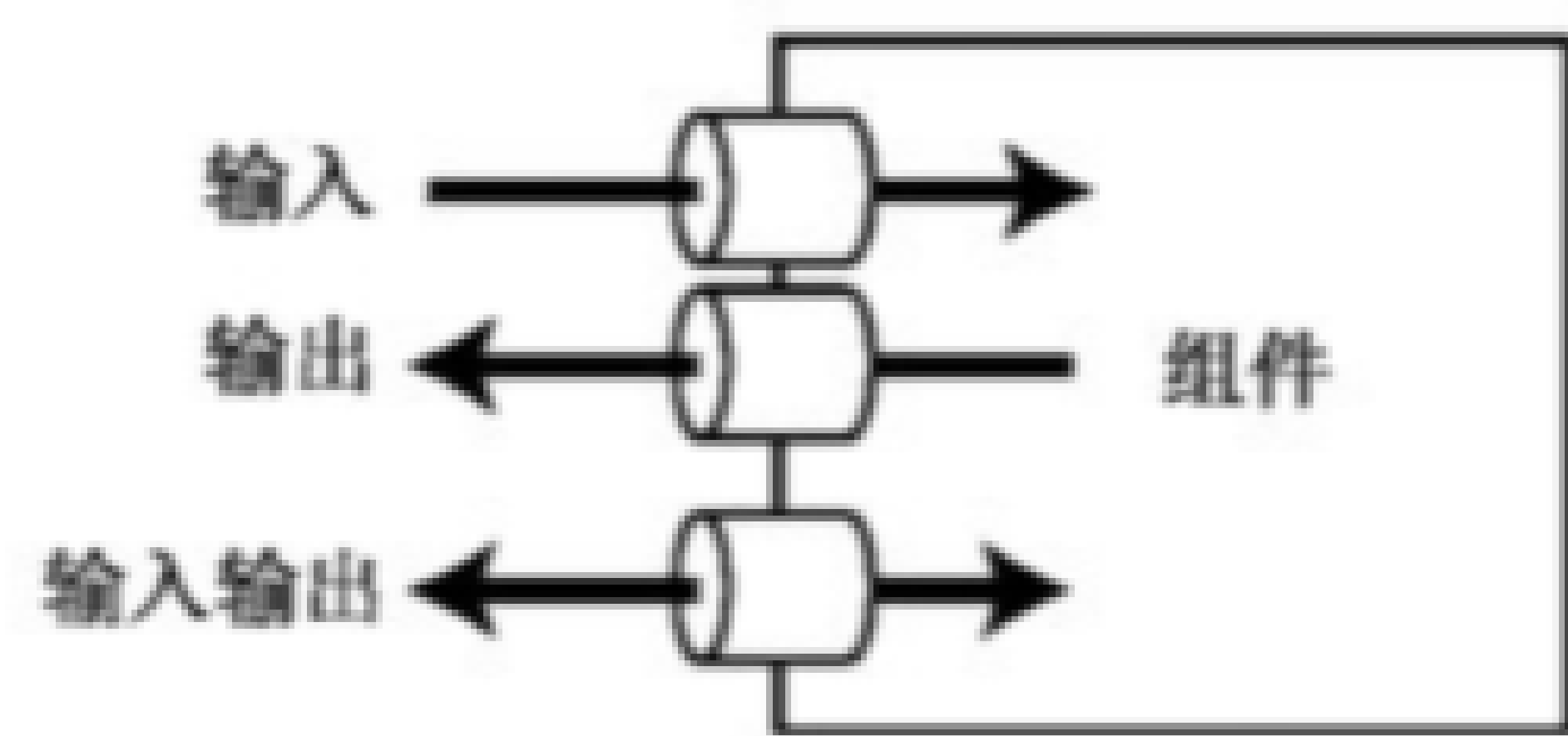


图 22 输入/输出端口与信号方向关系示意图

- h) type:此属性指定端口的端口类型。参数应为以下之一:
  - 1) power 电源网络端口;
  - 2) ground 接地网络端口;
  - 3) signal 信号网络用端口;
  - 4) floating 端口不应连接到任何网络;
  - 5) dontcare 端口没有任何逻辑意义,例如导热球;
  - 6) through 完全穿过组件的端口,即馈通(或直接连线);
  - 7) thermal 端口用于热网络,单位为开尔;
  - 8) thermal\_c 端口用于热网络,单位为摄氏度。

可不考虑 floating、dontcare 和 through 的信号方向。

8.2.5.5.3 语句组成

⟨port⟩语句包含以下内容：

⟨impedance⟩  
⟨delay⟩  
⟨voltage⟩  
⟨max\_current⟩  
⟨decap⟩  
⟨extensions⟩

8.2.5.5.4 例句

以下是⟨port⟩语句的示例。

```
⟨port id="A1" x="-12500" y="12500" direction="input"⟩  
  ⟨impedance typ="50"/⟩  
  ⟨delay typ="100"/⟩  
⟨/port⟩  
⟨port id="A2" x="-11500" y="12500" direction="output"⟩  
  ⟨impedance typ="50"/⟩  
  ⟨delay typ="100"/⟩  
⟨/port⟩
```

8.2.5.5.5 impedance 语句

8.2.5.5.5.1 概述

⟨impedance⟩语句定义端口的实际 I/O 阻抗。

⟨impedance typ="port\_impedance " /⟩

8.2.5.5.5.2 属性定义

⟨impedance⟩语句的属性如下：

typ: 此属性指定端口的实际负载阻抗, 阻抗单元由⟨unit⟩语句中的⟨impedance⟩语句定义。

8.2.5.5.6 delay 语句

8.2.5.5.6.1 概述

⟨delay⟩语句定义了实际的向后延迟和前向延迟, 它们是从输入端口到内部逻辑的延迟, 反之亦然。

⟨delay typ="port\_delay " /⟩

8.2.5.5.6.2 定义属性

⟨delay⟩语句的属性定义如下：

typ: 此属性指定端口的实际延迟, 延迟单元由⟨unit⟩语句中的⟨time⟩语句定义。



8.2.5.5.7 voltage 语句

8.2.5.5.7.1 概述

⟨voltage⟩语句定义了稳压 IC 或逻辑 IC 的额定输出电压。

```
⟨voltage
    typ="power_voltage"
    [ripple_rate="ripple_rate"]
/⟩
```

8.2.5.5.7.2 属性定义

⟨voltage⟩语句的属性定义如下。

- a) typ:该属性指定额定输出电压。对于稳压 IC,该属性表示输出电压引脚的电压。对于逻辑 IC,该属性表示输出引脚的信号电压。电压的单位是由⟨unit⟩语句中的⟨voltage⟩语句定义。
- b) ripple\_rate:该属性定义了稳压 IC 输出电压的纹波。

8.2.5.5.8 max\_current 语句

8.2.5.5.8.1 概述

⟨max\_current⟩语句定义了稳压 IC 或逻辑 IC 的最大输出电流。

```
⟨max_current typ="maximum_current_consumption"/⟩
```

8.2.5.5.8.2 属性定义

⟨max\_current⟩语句的属性定义如下：

Typ:此属性定义最大输出电流。对于稳压 IC,该属性表示在额定输出电压下的最大输出电流。在逻辑 IC 的情况下,该属性表示可以从输出端口输出的最大电流值。电流的单位是由⟨unit⟩语句中的⟨current⟩语句定义的。

8.2.5.5.9 dumping 语句

8.2.5.5.9.1 概述

⟨dumping⟩语句定义了端口的转储电阻器的放置范围和电阻值。

```
⟨dumping
    distance="distance_to_resistor"
    resistance="resistance"
/⟩
```

8.2.5.5.9.2 属性定义

⟨dumping⟩语句的属性定义如下。

- a) distance:此属性指定端口转储电阻的放置范围。指定的值是从端口到电阻的可接受的最大距离。距离的单位是由⟨unit⟩语句中的⟨distance⟩语句定义的。
- b) resistance:该属性指定转储电阻的电阻值。电阻的单位由⟨unit⟩语句中的⟨resistance⟩语句定义。

8.2.5.5.10 decap 语句

8.2.5.5.10.1 概述

〈decap〉语句定义了端口的去耦电容的放置范围和电容值。

```
〈decap
    distance="distance_to_capacitor"
    capacitance="capacitance"
/〉
```

8.2.5.5.10.2 属性定义

〈decap〉语句的属性定义如下。

- a) distance:该属性指定端口的去耦电容的放置范围。指定的值是从端口到去耦电容的可接受的最大距离。距离的单位由〈unit〉语句中的〈distance〉语句定义。
- b) Capacitor:该属性指定电容的电容值。电容的单位由〈unit〉语句中的〈capacitance〉语句定义。

8.2.5.6 portgroup 语句

8.2.5.6.1 概述

〈portgroup〉语句定义由同一文件中的其他语句引用的端口组。

```
〈portgroup
    name="port_group_name "
〉
    [〈moujoin/〉]
    [〈differential/〉]
    [〈ref_port〉 element]...
    [〈ref_portgroup〉 element]...
    [〈extensions〉 element]...
/〉
```

如果需要，一个端口可属于多个组，这些组可嵌套成其他组，可通过收集多个端口组来创建新的端口组。

8.2.5.6.2 定义属性

〈portgroup〉语句的属性定义如下：

name:此属性指定用于从其他属性和语句引用的端口组的名称，组名称在〈socket〉语句中是唯一的。

8.2.5.6.3 语句组成

〈portgroup〉语句包含以下内容：

```
〈mustjoin〉
〈differential〉
```

```
<ref_port>  
<ref_portgroup>  
<extensions>
```

8.2.5.6.4 例句

以下是<portgroup>语句的示例。

```
<portgroup name="ADD">  
    <ref_port name="AD0" />  
    <ref_port name="AD1" />  
    <ref_port name="AD2" />  
    <ref_port name="AD3" />  
</portgroup>  
  
<portgroup name="DIGITAL_GND">  
    <mustjoin/>  
    <ref_port name="CORE_GND" />  
    <ref_port name="IO_GND" />  
</portgroup>  
  
<portgroup name="ANALOG_GND">  
    <mustjoin/>  
    <ref_port id="A1" />  
    <ref_port id="A2" />  
</portgroup>  
  
<portgroup name="GND">  
    <ref_portgroup name="DIGITAL_GND" />  
    <ref_portgroup name="ALALOG_GND" />  
</portgroup/>
```

8.2.5.6.5 mustjoin 语句

如果<mustjoin/>包含在<port group>语句中,则组中的端口将连接在一起。

```
<mustjoin/>
```

8.2.5.6.6 differential 语句

如果<port\_group>语句中包含<differential/>语句,则该组中的端口为差分信号。

```
<differential/>
```

8.2.5.6.7 ref\_port 语句

8.2.5.6.7.1 概述

⟨ref\_port⟩语句关于成组的端口。

```
⟨ref_port
    { id="identifier_of_referenced_port " |
      name="name_of_referenced_port " }
    [polarity=" polarity "]
/⟩
```

使用名称(name)或标识符(id)端口应被指定,所引用的端口应在相同的⟨socket⟩语句中定义。

8.2.5.6.7.2 定义属性

⟨ref\_port⟩语句的属性定义如下。

- a) id:此属性指定组的预定义端口的标识符,指定的端口应定义在相同的⟨socket⟩语句中的⟨port⟩语句中,id 属性不应与 name 属性一起使用。
- b) name:此属性指定组的预定义端口的名称,指定的端口应定义在相同的⟨socket⟩语句中的⟨port⟩语句中,name 属性不应与 id 属性一起使用,如果存在多个同名端口,则这些端口属于同一组。
- c) polarity:此属性指定差分符号的极性。

NEGATIVE	负信号
POSITIVE	正信号

8.2.5.6.8 ref\_portgroup 语句

8.2.5.6.8.1 概述

端口组可以嵌套到另一个端口组中,⟨ref\_portgroup⟩语句引用另一个端口组。

```
⟨ref_portgroup
    name="name_of_referenced_port_group "
/⟩
```

引用的端口组应在相同的⟨socket⟩语句中定义。

8.2.5.6.8.2 定义属性

⟨ref\_portgroup⟩语句的属性定义如下：

name:此属性指定组成嵌套组的预定义端口组的名称,指定的端口组应在相同的⟨socket⟩语句中定义。

8.2.5.7 powerdomain\_group 语句

8.2.5.7.1 通则

⟨powerdomain\_group⟩语句定义了从组件端口进出的信号的电源域。

```
⟨powerdomain_group
    { port_name="name_of_referenced_port " |
      port_id="identifier_of_referenced_port " |
      group_name="name_of_referenced_port_group " }
    [min="minimum_voltage "]
    [typ="typical_voltage "]
    [max="maximum_voltage "]
    {pwr_port_name="name_of_referenced_power_port " |
      pwr_port_id="identifier_of_referenced_power_port " |
      pwr_port_name="name_of_referenced_power_port_group "}
    [pwr_min="minimum_powder_voltage "]
    [pwr_typ="typical_powder_voltage "]
    [pwr_max="maximum_powder_voltage "]
    {gnd_port_name="name_of_referenced_ground_port " |
      gnd_port_id="identifier_of_referenced_ground_port " |
      gnd_port_name="name_of_referenced_ground_port_group "}
    [gnd_min="minimum_ground_voltage "]
    [gnd_typ="typical_ground_voltage "]
    [gnd_max="maximum_ground_voltage "]
    [ripple_rate=" ripple_rate "]
>

    [⟨ref_portgroup⟩ element]
    [⟨ref_port⟩ element]
    [⟨extensions⟩ element]...
⟨/powerdomain_group⟩
```

例如，在模拟-数字混合设计的情况下，该语句指定构成电源域的模拟电源/接地和模拟信号，以便将它们与数字区分开来。端口应使用端口名(port\_name)、端口标识符(port\_id)或端口组名称(group\_name)指定，引用的端口或端口组应在相同的⟨socket⟩语句中定义。

8.2.5.7.2 定义属性

⟨powerdomain\_group⟩语句的属性定义如下。

- a) pwr\_port\_name:此属性定义电源的名称，或定义电源域的电压级别。在相同的⟨socket⟩语句中的⟨port⟩语句中定义引用的电源端口，端口类型为电源，pwr\_port\_name 属性不应与 pwr\_port\_id 和 pwr\_group\_name 属性一起使用。
- b) pwr\_port\_id:此属性定义电源端口的标识符，该标识符定义电源域的电压电平，在相同的⟨socket⟩语句中的⟨port⟩语句中定义引用的电源端口，端口类型为电源。port\_id 属性不应与

port\_name 和 group\_name 属性一起使用。

- c) pwr\_port\_group: 此属性定义电源域电压等级的端口组的名称,指定的端口组应在相同的<socket>语句中的<portgroup>语句中定义,属于该组端口类型应为电源,pwr\_group\_name 属性不应与 pwr\_port\_name 和 pwr\_port\_id 属性一起使用。
- d) gnd\_port\_name: 此属性定义接地域电压等级的端口组的名称,指定的端口组应在相同的<socket>语句中的<port>语句中定义,属于该组端口类型应为接地,gnd\_group\_name 属性不应与 gnd\_port\_id 和 gnd\_group\_name 属性一起使用。
- e) gnd\_port\_id: 此属性定义接地域电压等级的端口组的名称,指定的端口组应在相同的<socket>语句中的<port>语句中定义,属于该组端口类型应为接地,gnd\_group\_name 属性不应与 gnd\_port\_id 和 gnd\_group\_name 属性一起使用。
- f) gnd\_port\_group: 此属性定义接地域电压等级的端口组的名称,指定的端口组应在相同的<socket>语句中的<port>语句中定义,属于该组端口类型应为接地,gnd\_group\_name 属性不应与 gnd\_port\_id 和 gnd\_group\_name 属性一起使用。
- g) pwr\_typ: 此属性定义典型的电源端电压等级,电压扰动由最小和最大属性指定,电压单元由<unit>语句中的<voltage>语句定义。
- h) pwr\_min/max: 这些属性定义电源端电压扰动水平,pwr\_max 和 pwr\_min 属性分别为最大值和最小值电源端电压,电压单元由<unit>语句中的<voltage>语句定义。
- i) gnd\_typ: 这些属性定义了典型接地端电压等级。电压扰动由最小和最大属性指定,电压单元由<unit>语句中的<voltage>语句定义。
- j) gnd\_min/max: 这些属性定义了接地电压的扰动水平。gnd\_max 和 gnd\_min 属性分别为最大值和最小值接地电压。电压单位由<unit>语句中的<voltage>语句定义。
- k) ripple\_rate: 此属性定义了功率域的允许纹波速率。

为了向后兼容,不建议使用以下属性。在此文件的未来修订版中,将删除以下属性。

- a) potr\_name: 此属性定义电源域电压等级的电源和接地端口组的名称,指定的电源和接地端口组应在相同的<socket>语句中的<port>语句中定义,属于该组端口类型应为电源或接地,port\_name 属性不应与 port\_id 和 group\_name 属性一起使用。
- b) port\_id: 此属性定义电源域电压等级的电源和接地端口组的名称,指定的电源和接地端口组应在相同的<socket>语句中的<port>语句中定义,属于该组端口类型应为电源或接地,port\_id 属性不应与 port\_name 和 group\_name 属性一起使用。
- c) group\_name: 此属性定义电源域电压等级的端口组的名称,指定的端口组应在相同的<socket>语句中的<portgroup>语句中定义,属于该组端口类型应为电源或接地,group\_name 属性不应与 port\_name 和 port\_id 属性一起使用。
- d) typ: 此属性定义典型的电压等级,电压扰动由最小和最大属性指定,电压单元由<unit>语句中的<voltage>语句定义。
- e) min/max: 这些属性定义电压扰动水平,最大和最小属性分别是最大电压和最小电压,电压单元由<unit>语句中的<voltage>语句定义。

8.2.5.7.3 语句组成

<powerdomain\_group>语句包含以下内容:

<ref\_portgroup>  
<ref\_port>  
<extensions>

8.2.5.7.4 例句

以下是<powerdomain\_group>语句的一个示例。

```
<portgroup name="DDR_DIG">
  <ref_port name="PL1" />
  <ref_port name="PL2" />
  <ref_port name="PL3" />
  <ref_port name="PL4" />
</portgroup>
<powerdomain_group  pwr_port_name="DCDC"  pwr_typ= "2.5"
                    gnd_port_name="PGND"  gnd_typ = "0.0">
  <ref_portgroup name="DDR_DIG" />
</portdomain_group>
```

8.2.5.7.5 ref\_portgroup 语句

8.2.5.7.5.1 概述

<ref\_portgroup>语句引用构成电源域的端口组。

```
<ref_portgroup
  name="name_of_referenced_port_group "
/>
```

引用的端口组应在相同的<socket>语句中定义。

8.2.5.7.5.2 定义属性

<ref\_portgroup>语句的属性定义如下：

name:此属性定义组成电源域的预定义信号端口组的名称,包含在同一组中的端口属于同一电源域,指定的端口组应在相同的<socket>语句中定义。

8.2.5.7.6 ref\_port 语句

8.2.5.7.6.1 概述

<ref\_port>语句引用构成电源域的信号端口。

```
<ref_port
  { id="identifier_of_referenced_port " |
    name="name_of_referenced_port " }
/>
```

端口应使用名称(name)或标识符(id)指定,引用的端口应在相同的<socket>语句中定义。

8.2.5.7.6.2 定义属性

- <ref\_port>语句的属性定义如下。
  - a) Id:此属性定义组成电源域的预定义信号端口的标识符。指定的信号端口应定义在<socket>语句中的<port>语句中,id 属性不应与 name 属性一起使用。
  - b) name:此属性定义组成电源域的预定义信号端口的名称。指定的信号端口应定义在相同的<socket>语句的<port>语句中,name 属性不应与 id 属性一起使用。如果存在多个同名端口,则这些端口属于同一电源域。

8.2.5.8 swappable\_port 语句

8.2.5.8.1 概述

<swappable\_port>语句定义可交换端口集,例如 DDR 3 数据总线字节。

```

<swappable_port>
    <ref_port> element
    {<ref_port> element}...
    [<extensions> element]...
</swappable_port>

```

<swable-port>语句包含两个或多个<ref\_port>语句。连接到<ref\_port>语句定义的可交换端口的网络可以在连接中相互替换。例如,如果图 23 所示的可互换的 CK\_N 和 CK\_P 分别连接到 n1 和 n2,则可交换 CK\_N 和 CK\_P 的连接。

8.2.5.8.2 语句组成

<swappable\_port>语句包含以下内容:

```

<ref_port>
<extensions>

```

8.2.5.8.3 例句

图 23 所示的可交换端口由以下代码表示。

```

<swappable_port>
    <ref_port name="CK_N" />
    <ref_port name="CK_P" />
</swappable_port>

```

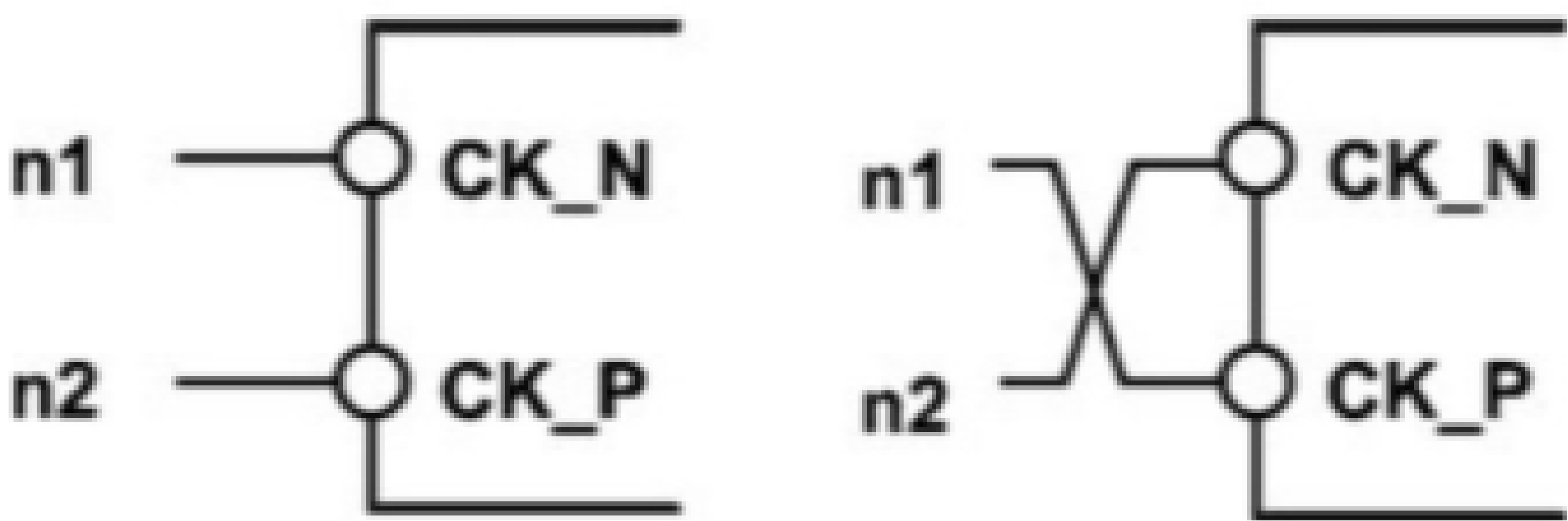


图 23 可交换端口示例



8.2.5.8.4 ref\_port 语句

8.2.5.8.4.1 通则

⟨ref\_port⟩语句引用构成可交换端口组的端口。

```
⟨ref_port
    { id="identifier_of_referenced_port " |
      name="name_of_reference_port " }
/⟩
```

端口应使用名称(name)或标识符(id)属性定义,引用的端口组应在相同的⟨socket⟩语句中定义。

8.2.5.8.4.2 定义属性

⟨ref\_port⟩语句的属性如下:

- a) id:此属性定义组成一组可交换预定义端口的标识符,指定的端口应在相同的⟨socket⟩语句中的⟨port⟩语句定义,id 属性不应与 name 属性一起使用;
- b) name:此属性定义组成一组可交换预定义端口的名称,指定的端口应在相同的⟨socket⟩语句的⟨port⟩语句中定义,name 属性不应与 id 属性一起使用。

8.2.5.9 port\_assignment 语句

8.2.5.9.1 概述

⟨port\_assignment⟩语句用于定义要分配给物理端口的信号。

```
⟨port_assignment⟩
    {⟨ref_port_by_id⟩element}...
    {⟨ref_port_by_name⟩element}....
    [⟨extensions⟩element]...
⟨/port_assignment⟩
```

物理端口由具有 id 属性的⟨port⟩语句定义。另一方面,分配给物理端口的信号由⟨port⟩语句中的 name 属性定义。没有⟨name⟩属性的⟨port⟩语句表示尚未分配信号。没有 id 属性的⟨port⟩语句表示尚未为信号定义物理端口。⟨port\_assignment⟩由以下两个语句组成:

```
⟨ref_port_by_id⟩
⟨ref_port_by_name⟩
```

⟨ref\_port\_by\_id⟩语句通过 id 属性引用端口,从而引用物理端口。⟨ref\_port\_by\_name⟩语句按名称引用端口属性,从而引用信号。⟨port\_assignment⟩的含义是,能将被引用的信号分配给该物理端口,不能将被引用的信号分配给其他物理端口,也不能将其他信号分配给被引用的物理端口。

8.2.5.9.2 ref\_port\_by\_id 语句

8.2.5.9.2.1 通则

⟨ref\_port\_by\_id⟩语句定义尚未为其分配信号的物理端口。

⟨ref\_port\_by\_id id="identifier\_of\_referenced\_port"/⟩

此语句通过 id 属性引用相同⟨socket⟩语句中的唯一⟨port⟩语句。如果所引用的⟨port⟩语句已被赋值给一个信号,则该赋值应视为暂定。

8.2.5.9.2.2 属性定义

⟨ref\_port\_by\_id⟩语句的属性如下:

Id:此属性指定预定义的标识符。指定的端口应在相同⟨socket⟩语句中的⟨port⟩语句中定义。

8.2.5.9.3 ref\_port\_by\_name 语句

8.2.5.9.3.1 通则

⟨ref\_port\_by\_name⟩语句定义尚未分配给物理端口的信号。

⟨ref\_port\_by\_name name="name\_of\_reference\_signal"/⟩

此语句通过指定信号名称来引用相同⟨socket⟩语句中的一个或多个⟨port⟩语句。如果信号已分配给⟨socket⟩语句中的一个或多个物理端口,则该分配应视为暂定。

8.2.5.9.3.2 属性定义

⟨ref\_port\_by\_name⟩语句的属性如下:

name:此属性指定了一个信号名称,该信号名称应在相同的⟨socket⟩语句的⟨port⟩语句中定义。

8.2.5.9.4 示例

```
⟨socket name="example_socket1"⟩
  ⟨port id="A1"x="0"y="0"shape_id="port_shapel"
    name="sig1"direction="in"/⟩
  ⟨port id="A2"x="1"y="0"shape_id="port_shapel"
    name="sig3"direction="out"/⟩
  ⟨port id="A3"x="1"y="1"shape_id="port_shapel"
    name="sig2"direction="in"/⟩
  ⟨port id="A4"x="0"y="1"shape_id="port_shapel"
    name="sig4"direction="out"/⟩
  ⟨port_assignment⟩
    ⟨ref_port_by_id id="A1"/⟩
    ⟨ref_port_by_id id="A2"/⟩
    ⟨ref_port_by_name name="sig1"/⟩
    ⟨ref_port_by_name name="sig3"/⟩
  ⟨/port_assignment⟩
```

```
<port_assignment>
  <ref_port_by_id id="A3"/>
  <ref_port_by_id id="A4"/>
  <ref_port_by_name name="sig2"/>
  <ref_port_by_name name="sig4"/>
</port_assignment>
```

8.2.5.10 swappable\_group 语句

8.2.5.10.1 通则

<swappable\_group>语句定义可交换端口组的集合。

```
<swappable_group>
  <ref_portgroup> element
  {<ref_portgroup> element}...
  [<extensions> element]...
</swappable_group>
```

同一<swappable\_group>中的端口组可由组单元进行交换(参见图 24 和图 25)。<swappable\_group>包含两个或多个<ref\_portgroup>语句,连接到<ref\_portgroup>语句定义的可交换端口组的网络可按组单元在连接中相互重新放置。如果所有<ref\_portgroup>语句中的端口计数匹配,则应在替换中保持网络的顺序。

8.2.5.10.2 语句组成

<swappable\_group>语句包含以下内容:

```
<ref_portgroup>
<extensions>
```

8.2.5.10.3 例句

图 24 所示的可交换组由以下代码表示。

```
<portgroup name="R_CHANNEL">
  <ref_port name="RAD1" />
  <ref_port name="RAD2" />
  <ref_port name="RAD3" />
</portgroup>
<portgroup name="L_CHANNEL" />
  <ref_port name="LAD1" />
  <ref_port name="LAD2" />
  <ref_port name="LAD3" />
</portgroup>

<swappable_group>
  <ref_portgroup name="R_CHANNEL" />
```

```
<ref_portgroup name="L_CHANNEL" />
</swappable_group>
```

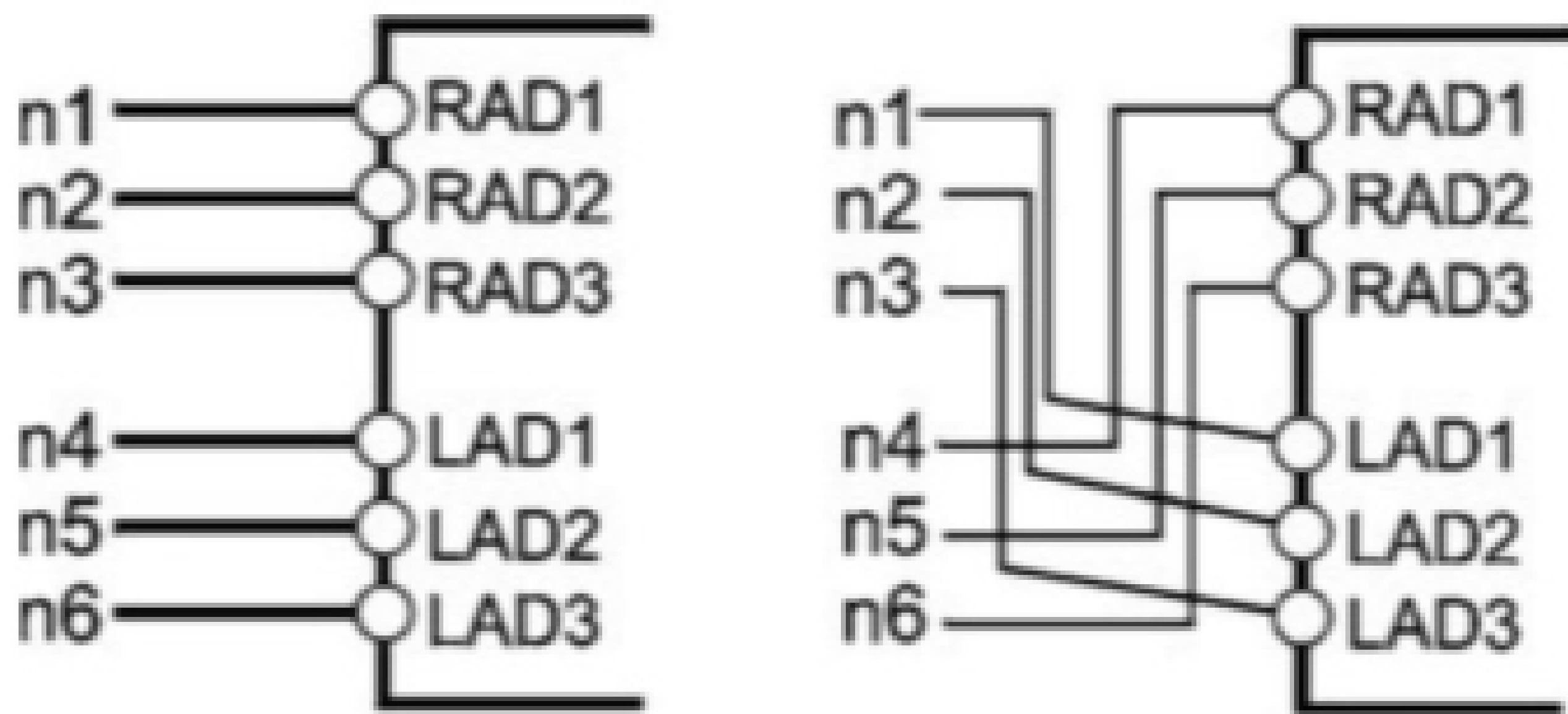


图 24 可交换组的示例

图 25 所示的可交换组由以下代码表示。

```
<portgroup name="RCHA_1">
  <ref_port name="RAD1" />
  <ref_port name="RAD2" />
</portgroup>
<portgroup name="RCHA_2">
  <ref_port name="RAD3" />
  <ref_port name="RAD3" />
</portgroup>
<portgroup name="R_CHANNEL" />
  <ref_portgroup="RCHA_1" />
  <ref_portgroup="RCHA_2" />
</portgroup>
<portgroup name="L_CHANNEL" />
  <ref_port name="LAD1" />
  <ref_port name="LAD2" />
  <ref_port name="LAD3" />
  <ref_port name="LAD4" />
</portgroup>
<swappable_group>
  <ref_portgroup name="R_CHANNEL" />
  <ref_portgroup name="L_CHANNEL" />
</swappable_group>
```

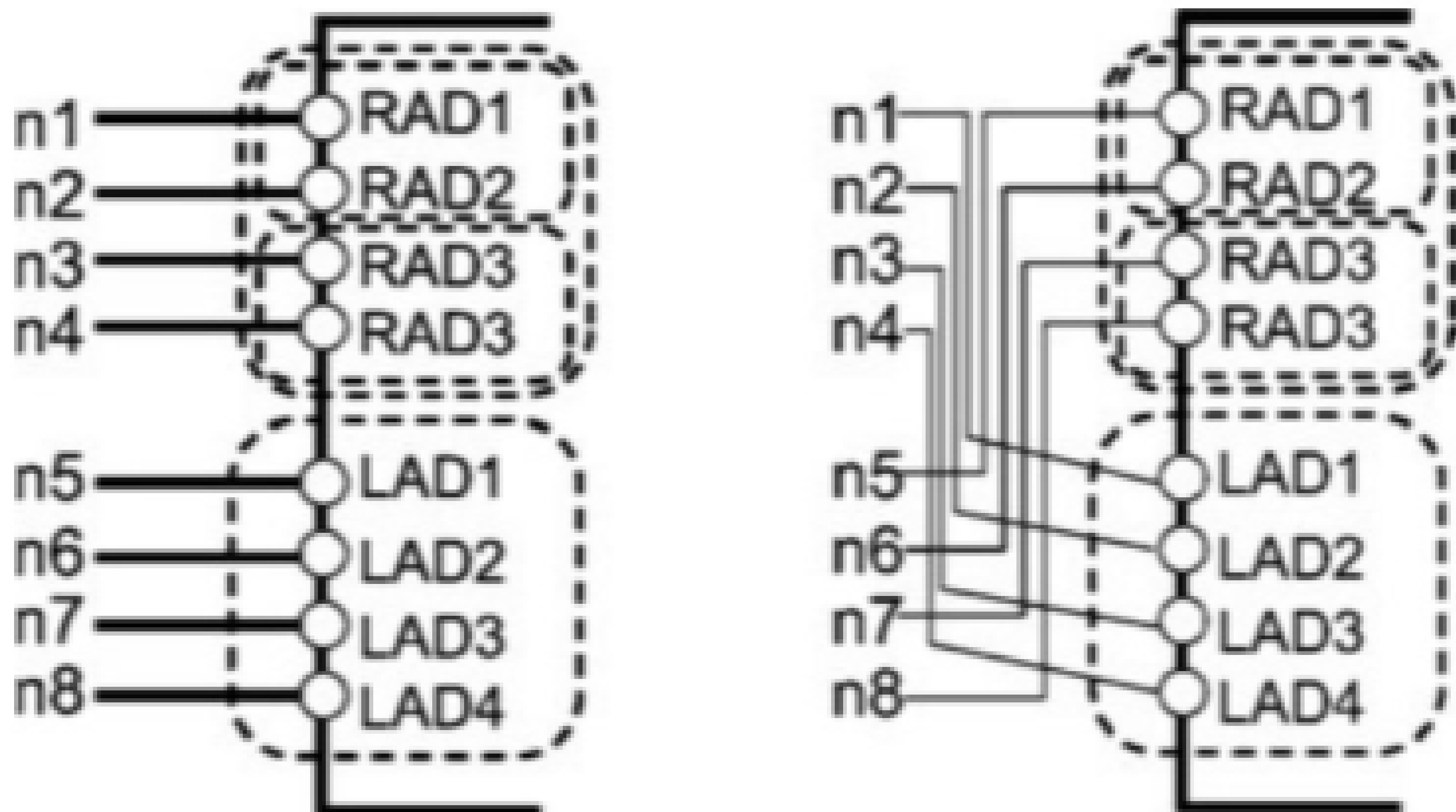


图 25 可交换组的示例

8.2.5.10.4 ref\_portgroup 语句

8.2.5.10.4.1 概述

⟨ref\_portgroup⟩语句引用两个或多个端口组。

```
⟨ref_portgroup
    name="group_name_of_reference_port "
/⟩
```

属于引用组的端口可以交换。例如,如果以下两个端口组 AGRP 和 BGRP 是可交换组,则端口 A0 和 B0、A1 和 B1 以及 A2 和 B2 分别可互换。

```
⟨portgroup name="AGRP"⟩
    ⟨ref_port name="A0" /⟩
    ⟨ref_port name="A1" /⟩
    ⟨ref_port name="A2" /⟩
⟨/portgroup⟩
```

```
⟨portgroup name="BGRP"⟩
    ⟨ref_port name="B0" /⟩
    ⟨ref_port name="B1" /⟩
    ⟨ref_port name="B2" /⟩
⟨/portgroup⟩
```

8.2.5.10.4.2 定义属性

⟨ref\_portgroup⟩语句的属性定义如下:

name:此属性指定组成一组可交换预定义端口组的名称,指定的端口组应在相同的⟨socket⟩语句中定义。

8.2.5.11 frequency 语句

8.2.5.11.1 通则

⟨frequency⟩语句定义了移动组件端口的信号的最小、典型和最大工作频率。

```
⟨frequency
    { port_name="name_of_reference_port " |
      port_id="identifier_of_reference_port " |
      group_name="identifier_of_reference_port_group " }
    min="minimum_frequency "
    [typ="typical_frequency "]
    max="maximum_frequency "
/⟩

[⟨extensions⟩element]...
```

</frequency>

端口应使用端口名(Port\_name)、端口标识符(Port\_Id)或组名(Group\_name)定义。

8.2.5.11.2 定义属性

<frequency>语句的属性定义如下。

- a) port\_name:此属性指定用于输入和输出规定的工作频率信号的预定义端口的名称,指定的端口应定义在相同的<socket>语句的<port>语句中,port\_name 属性不应与 port\_id 或 group\_name 属性一起使用。
- b) port\_id:此属性指定用于输入和输出规定的工作频率的信号的预定义端口的标识符,指定的端口应定义在相同的<socket>语句的<port>语句中,port\_id 属性不应与 port\_name 或 group\_name 属性一起使用。
- c) group\_name:此属性指定包括用于输入和输出规定的工作频率信号端口组的组名称,端口组包含的端口以相同的频率工作。指定的端口组应在相同的<socket>语句的<portgroup>语句中定义,group\_name 属性不应与 port\_name 或 port\_id 属性一起使用。
- d) typ:此属性指定典型频率,频率扰动由 min 和 max 属性定义,频率单元由<unit>语句中的<frequency>语句定义。
- e) min/max:此属性定义频率扰动。max 和 min 属性分别是最大频率和最小频率,频率单元由<unit>语句中的<frequency>语句定义。

8.2.5.11.3 例句

以下是<frequency>语句的一个示例。

<frequency port\_name="FKBCLK" min="50" max="55" />

8.2.5.12 constraint 语句

8.2.5.12.1 概述

<constraint>语句定义设计约束,例如偏移的限制。

<constraint>  
    [<impedance> element]  
    [<delay> element]  
    [<skew> element]  
    [<guard\_shield> element]  
    [<extensions> element]  
</constraint>

<constraint>语句由一个或多个<impedance>、<delay>、<skew>和<guard\_shield>语句组成。

8.2.5.12.2 语句组成

<constraint>语句包含以下内容:

<impedance>  
<delay>  
<skew>  
<guard\_shield>  
<extensions>

8.2.5.12.3 impedance 语句

8.2.5.12.3.1 概述

<impedance>语句定义要求阻抗匹配的端口的最小、典型和最大特性阻抗。

```
<impedance
    { port_name="name_of_reference_port " |
      port_id="identifier_of_referenced_port " |
      group_name="name_of_reference_port_group " }
    [type="impedance_type "]
    [min="minimum_impedance "]
    typ="typical_impedance "
    [max="maximum_impedance "]
/>>
```

端口应使用端口名(port\_name)、端口标识符(port\_id)或(group\_name)属性定义。

8.2.5.12.3.2 定义属性

<impedance>语句的属性定义如下：

- a) port\_name:此属性指定用于输入和输出要求阻抗匹配的信号的预定义端口的名称,指定的端口应定义在相同的<socket>语句的<port>语句中,port\_name 属性不应与 port\_id 或 group\_name 属性一起使用。
- b) port\_id:此属性指定用于输入和输出要求阻抗匹配的信号的预定义端口的标识符,指定的端口应定义在相同的<socket>语句的<port>语句中,port\_id 属性不应与 port\_name 或 group\_name 属性一起使用。
- c) group\_name:此属性指定包括用于输入和输出要求阻抗匹配的信号的端口组的组名称,指定的端口组应在相同的<socket>语句的<portgroup>语句中定义,group\_name 属性不应与 port\_name 或 port\_id 属性一起使用。
- d) type:此属性指定特征阻抗的类型,参数应为下列之一:
  - single : 单端阻抗;
  - differential : 差分阻抗;
  - common : 共模阻抗。

如果未定义类型属性,则设置 single 为默认值。

- e) typ:此属性定义典型的阻抗值,阻抗扰动由最小和最大属性指定,阻抗单位由<unit>语句中的<impedance>语句定义。
- f) min/max:此属性指定阻抗扰动,最大和最小属性分别是最大和最小阻抗值,阻抗单位由<unit>语句中的<impedance>定义。

8.2.5.12.3.3 例句

以下是<impedance>语句的一个示例。

```
<portgroup name="CK">
  <ref_port port_name="CK_N" />
  <ref_port port_name="CK_P" />
</portgroup>
<constraint>
  <impedance port_name="DQ1" type="single" min="45" typ="50" max="55" />
  <impedance port_id="A1" type="single" typ="50" />
  <impedance group_name="CK" type="differential" typ="100" />
</constraint>
```

8.2.5.12.4 delay 语句

8.2.5.12.4.1 定义属性

<delay>语句的属性定义如下：

- a) port\_name:此属性指定用于输入和输出要求定时约束的信号的定义端口的名称,指定的端口应定义在相同的<socket>语句的<port>语句中,port\_name 属性不应与 port\_id 或 group\_name 属性一起使用。
- b) port\_id:此属性指定用于输入和输出要求定时约束的信号的定义端口的标识符,指定的端口应定义在相同的<socket>语句的<port>语句中,port\_id 属性不应与 port\_name 或 group\_name 属性一起使用。
- c) group\_name:此属性指定包括用于输入和输出要求定时约束的信号端口组的组名称,指定的端口组应在相同的<socket>语句的<portgroup>语句中定义,group\_name 属性不应与 port\_name 或 port\_id 属性一起使用。
- d) typ:此属性定义典型的延迟值,延迟时间扰动由最小和最大属性指定,延迟单位由<unit>语句中的<time>语句定义。
- e) min/max:此属性定义延迟时间扰动,max 和 min 属性分别是最大和最小延迟值,延迟单位由<unit>语句中的<time>或<distance>语句定义。

8.2.5.12.4.2 概述

<delay>语句定义要求定时约束端口的最小、典型和最大延迟值。

```
<delay
  { port_name="name_of_referenced_port " |
    port_id="identifier_of_referenced_port " |
    group_name="name_of_referenced_port_group "}
  [min="minimum_delay "]
  typ="typical_delay "
  [max="maximum_delay "]
/>
```



端口将使用端口名(port\_name)、端口标识符(port\_id)或组名(group\_name)属性定义。

8.2.5.12.4.3 例句

以下是<delay>语句的一个示例。

```
<portgroup name="BUS">
  <ref_port port_name="SIG1" />
  <ref_port port_name="SIG2" />
  <ref_port port_name="SIG3" />
</portgroup>
<constraint>
  <delay port_name="CLK" min="10" typ="12" max="15" />
  <delay port_id="D3" typ="5" />
  <delay group_name="BUS" min="50" typ="55" max="60"/>
</constraint>
```

8.2.5.12.5 skew 语句

8.2.5.12.5.1 概述

<skew>语句为从指定端口输入和输出的信号定义了偏移约束。

```
<skew
  { port_name="name_of_port " |
    port_id="identifier_of_port " |
    group_name="name_of_port_group " }
  [ reference_port_name="name_of_referenced_port " |
    reference_port_id="identifier_of_referenced_port " ]
  [min="minimum_time "]
  [max="maximum_time "]
/>
```

端口应使用端口名(port\_name)、端口标识符(port\_id)或组名(group\_name)属性定义。

当指定参考信号时,偏离约束由最大时间(max) 和最小时间 (min)定义。在这种情况下,指定的最大和最小时间是基于从指定的参考端口输入或输出的参考信号的传输时间(reference\_port\_name, reference\_port\_id)。具有偏移约束的信号的传输时间需要在最小到最大时间内结束(参见图 26)。

```
<portgroup name="skewset1">
  <ref_port name="port1"/>
  <ref_port name="port2"/>
  <ref_port name="port3"/>
</portgroup>
<skew group_name="skewset1"reference_port_name="port0"
  min="-5" max="10"/>
```

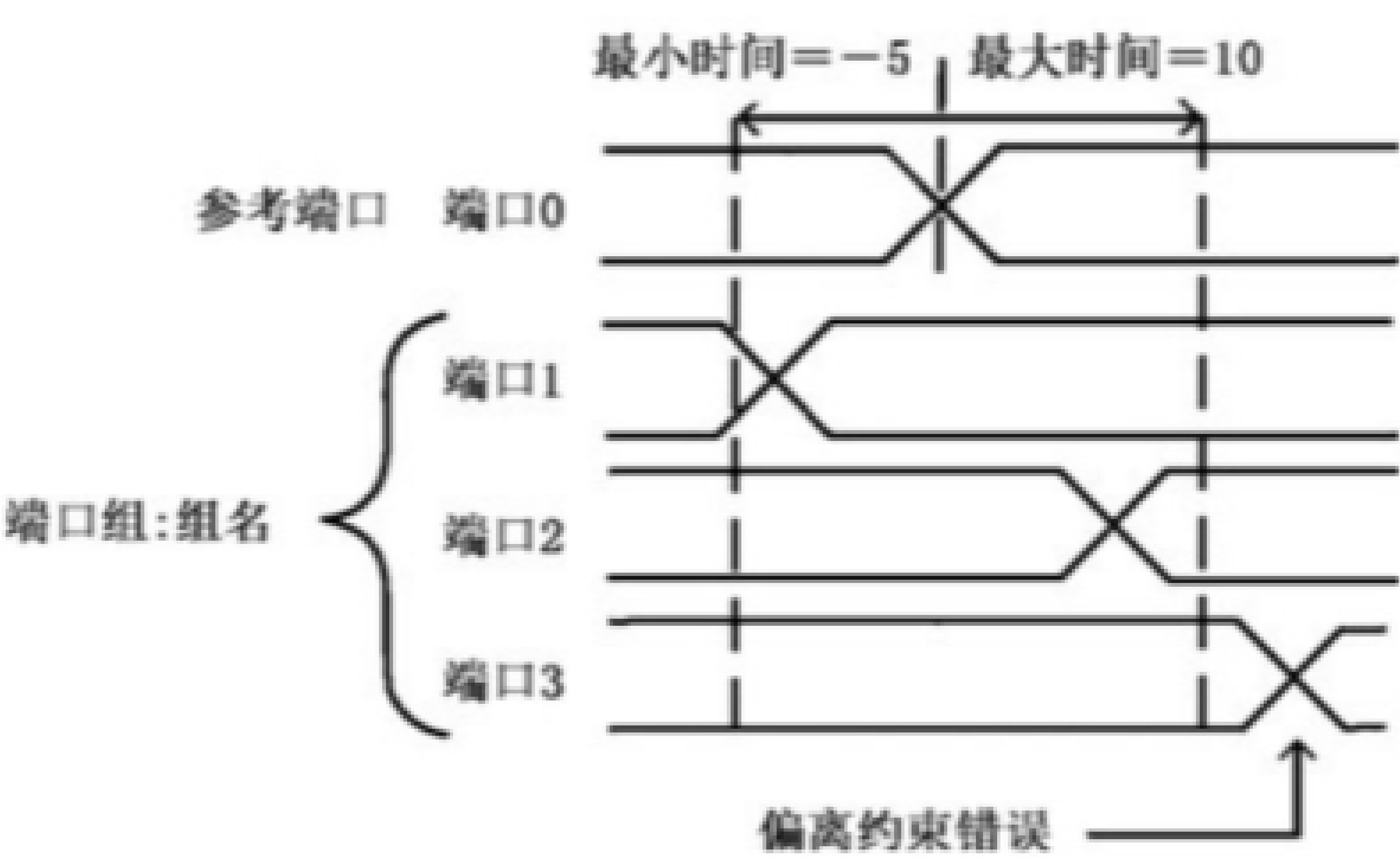


图 26 带有 min 和 max 属性的偏移约束示例

当未指定参考信号时，偏移约束由最大时间(max)或最小时间(min)定义。如果定义了最大时间,则使用传输时间最快的信号作为参考信号(见图 27)。

```
<portgroup name="skewset2">
  <ref_port name="port4"/>
  <ref_port name="port5"/>
  <ref_port name="port6"/>
</portgroup>
<skew group_name="skewset2" max="5"/>
```

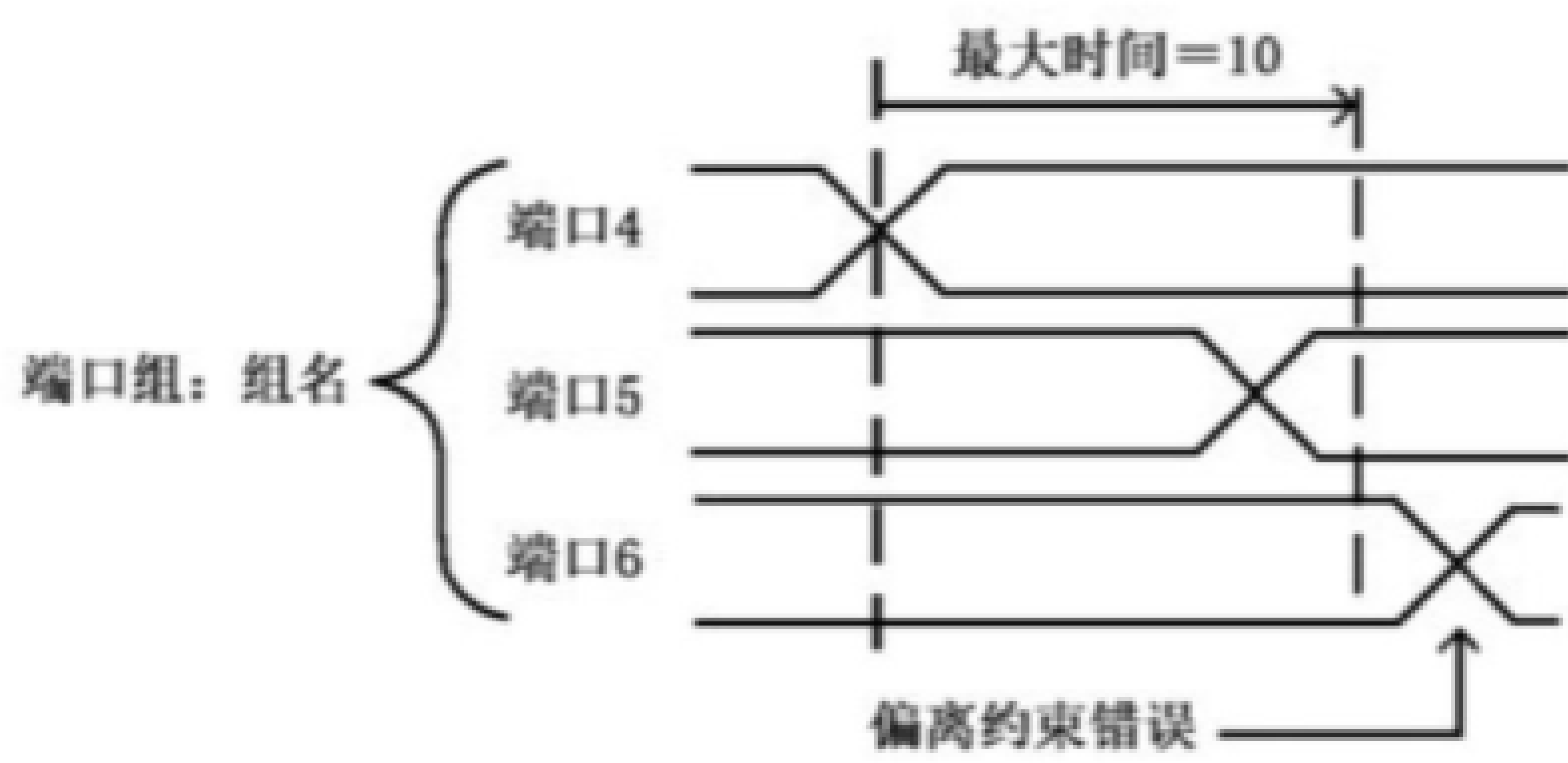


图 27 只具有 max 属性的偏移约束示例

如果只定义了最小时间(min),则使用传输时间最慢的信号作为参考信号(参见图 28)。

```
<portgroup name="skewset3">
  <ref_port name="port7"/>
  <ref_port name="port8"/>
  <ref_port name="port9"/>
</portgroup>
<skew group_name="skewset3" min="-10"/>
```

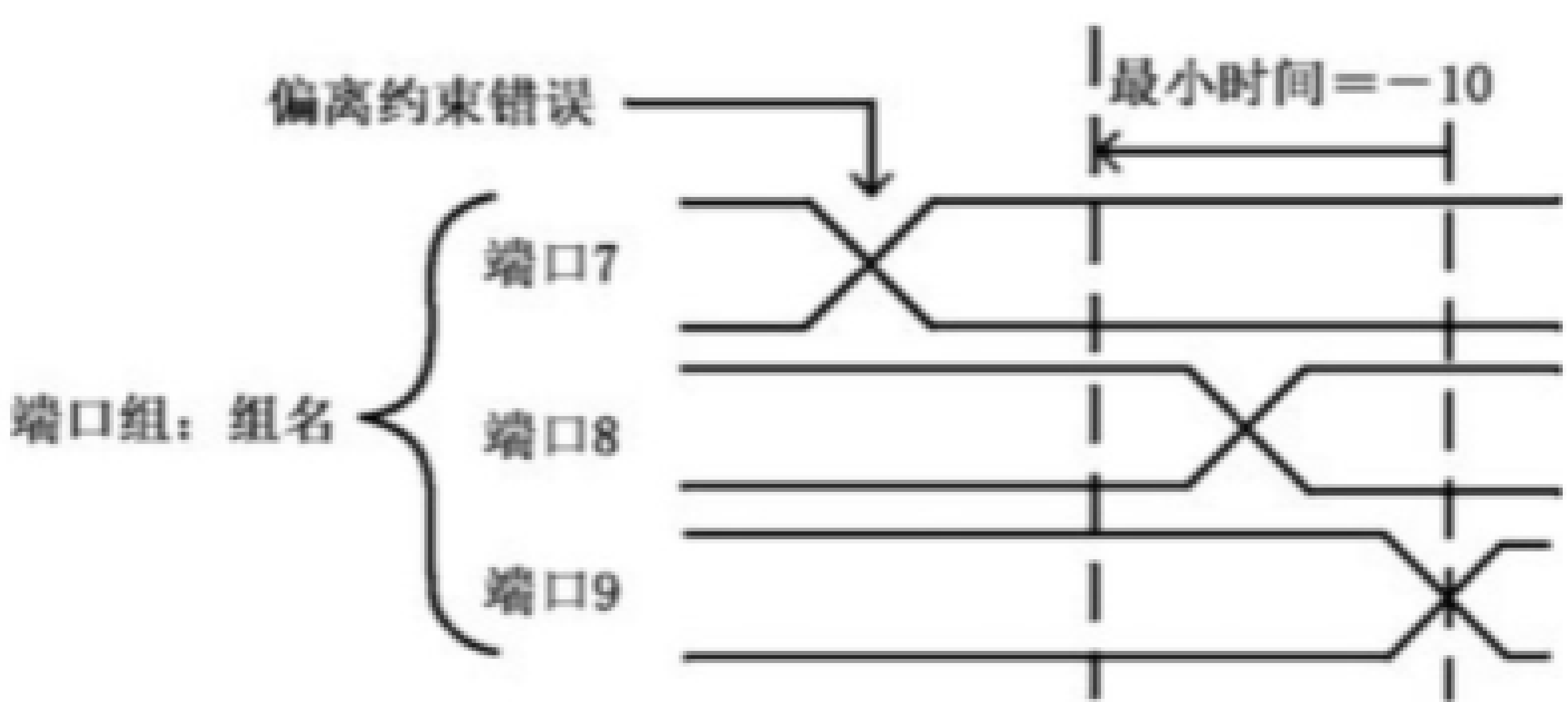


图 28 只具有 min 属性的偏移约束示例

8.2.5.12.5.2 定义属性

⟨skew⟩语句的属性定义如下。

- a) port\_name:此属性指定用于输入和输出要求偏移约束的信号的定义端口的名称,指定的端口应定义在相同的⟨socket⟩语句的⟨port⟩语句中,port\_name 属性不应与 port\_id 或 group\_name 属性一起使用。
- b) port\_id:此属性指定用于输入和输出要求偏移约束的信号定义端口的标识符,指定的端口应定义在相同的⟨socket⟩语句的⟨port⟩语句中,port\_id 属性不应与 port\_name 或 group\_name 属性一起使用。
- c) group\_name:此属性指定包括用于输入和输出要求偏移约束的信号端口的组名称,指定的端口组应在相同的⟨socket⟩语句的⟨portgroup⟩语句中定义,group\_name 属性不应与 port\_name 或 port\_id 属性一起使用。
- d) reference\_port\_name:此属性指定用于输入和输出偏移约束的引用信号的定义端口的名称,指定的端口应定义在相同的⟨socket⟩语句的⟨port⟩语句中,reference\_port\_name 属性不应与 reference\_port\_id 属性一起使用。
- e) reference\_port\_id:此属性指定用于输入和输出偏移约束的引用信号定义端口的标识符,指定的端口应定义在相同的⟨socket⟩语句的⟨port⟩语句中,reference\_port\_name 属性不应与 reference\_port\_id 属性一起使用。
- f) min/max:max 和 min 属性分别定义最大和最小偏斜时间,最大属性设置为正值,最小属性设置为负值,当参考信号由 reference\_port\_name 或 reference\_port\_id 指定时,min 或 max 属性应一起使用;当未指定参考信号时,可以使用 min 或 max 属性之一。最小和最大偏斜时间的单位由⟨unit⟩语句中的⟨time⟩语句定义。

8.2.5.12.5.3 例句

假设半导体设计者需要设计具有差分偏移约束的芯片,以下代码是由半导体设计者提供的 10 ps 的差分偏移约束的示例,封装设计者根据此约束条件设置差分信号。

```
⟨portgroup name="CK"⟩
  ⟨ref_port name="CKP" /⟩
  ⟨ref_port name="CKN" /⟩
⟨/portgroup⟩
⟨skew group_name="CK" max="10" /⟩
```

如果封装设计者以 100 ps 的延迟设置 CKP,以 95 ps 的延迟设置 CKN,则印制板设计者基于 CKP 的最小和最大偏移约束分别为-5 ps 和 15 ps。下面的代码和图 29 是印制板设计人员偏移约束的示例。

```
⟨portgroup name="CK"⟩
  ⟨ref_port name="CKP" /⟩
  ⟨ref_port name="CKN" /⟩
⟨/portgroup⟩
⟨skew group_name="CK" reference_port_name="CKP" min="-5" max="15" /⟩
```

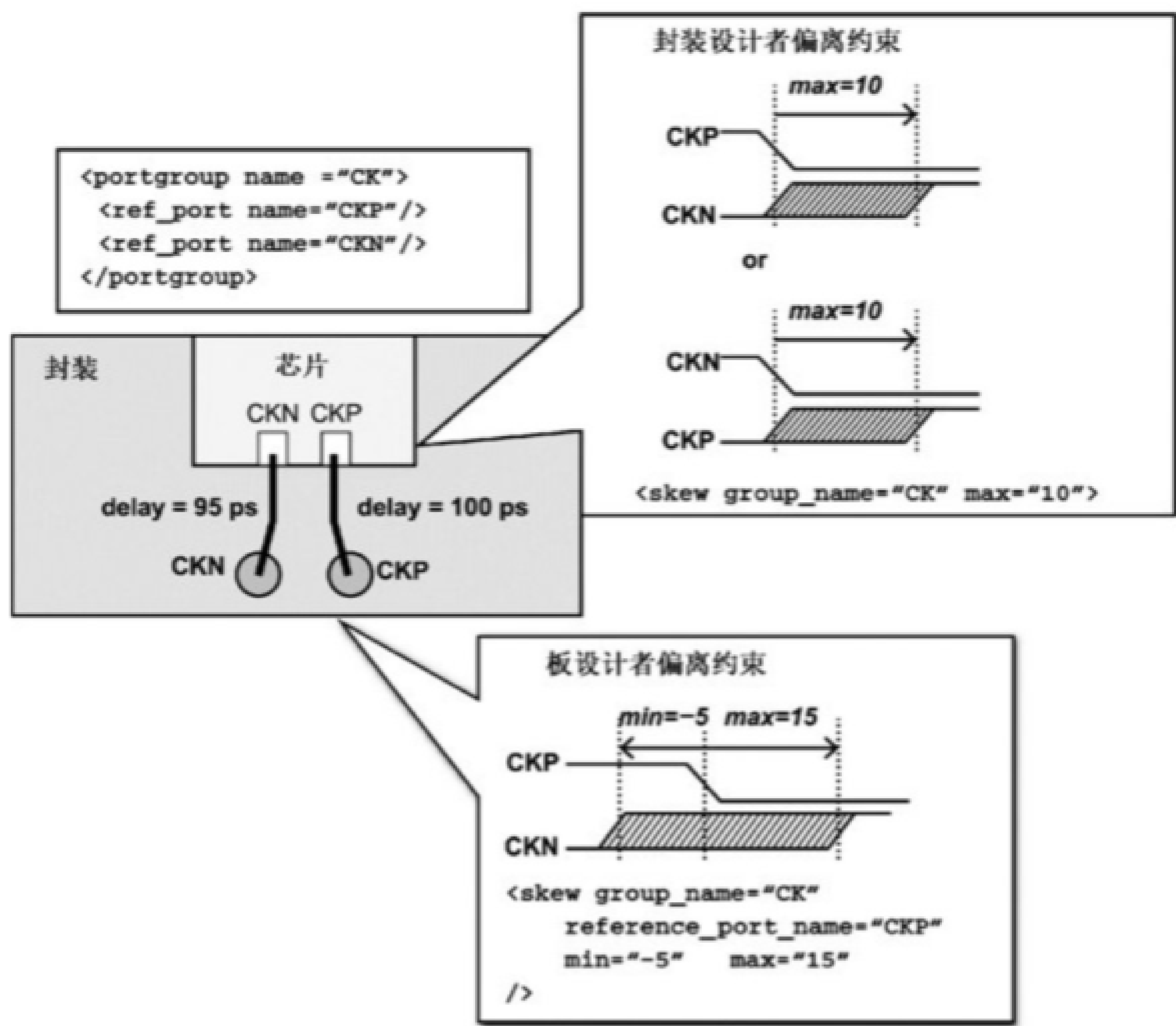


图 29 差分时钟的偏移约束示例

8.2.5.12.6 guard\_shield 语句

8.2.5.12.6.1 概述

〈guard\_shield〉语句定义了用于屏蔽的信号,以及需要屏蔽的信号。

```
<guard_shield
{ port_name="name_of_port_requiring_shield " |
port_id="identifier_of_port_requiring_shield " |
group_name="name_of_port_group_requiring_shield " }
{ shieldnet_port_name="name_of_port_using_shield" |
shieldnet_port_id="identifier_of_port_using_shield " |
shieldnet_port_group_name="name_of_port_group_using_shield " }
/>
```

例如,在模拟数字混合设计的情况下,最好用模拟地屏蔽模拟信号,guard\_shield 语句用于定义屏蔽信号的组合。

8.2.5.12.6.2 定义属性

〈guard\_shield〉语句的属性定义如下。

- a) port\_name:此属性指定用于输入和输出要求屏蔽的信号的定义端口的名称,指定的端口应定义在相同的〈socket〉语句的〈port〉语句中,port\_name 属性不应与 port\_id 或 group\_name 属性一起使用。
- b) port\_id:此属性指定用于输入和输出要求屏蔽的信号预定义端口的标识符,指定的端口应定

- 又在相同的<socket>语句的<port>语句中。port\_id 属性不应与 port\_name 或 group\_name 属性一起使用。
- c) group\_name:此属性指定包括用于输入和输出要求屏蔽的信号端口的组名称,指定的端口组应在相同的<socket>语句中的<portgroup>语句中定义。如果存在多个同名端口,则应屏蔽从这些端口输入或输出的信号,group\_name 属性不应与 port\_name 或 port\_id 属性一起使用。
  - d) shieldnet\_port\_name:此属性指定用于输入和输出用于屏蔽的信号的定义端口的名称,指定的端口应定义在<port>语句中的<socket>语句中,shieldnet\_port\_name 属性不应与 shieldnet\_port\_id 或 shieldnet\_port\_group\_name 属性一起使用。
  - e) shieldnet\_port\_id:此属性指定预定义端口的标识符,用于输入和输出屏蔽的信号,指定的端口应定义在<port>语句中的<socket>语句中,shieldnet\_port\_id 属性不应与 shieldnet\_port\_name 或 shieldnet\_port\_group\_name 属性一起使用。
  - f) shieldnet\_port\_group\_name:此属性指定组的名称,该组包括用于输入和输出屏蔽的信号端口。指定的端口组应在<socket>语句中的<portgroup>语句中定义,如果存在多个同名端口,则可使用从这些端口输入或输出的任何信号进行屏蔽,shieldnet\_port\_group\_name 属性不应与 shieldnet\_port\_name 或 shieldnet\_port\_id 属性一起使用。

8.2.5.12.6.3 例句

图 30 中的保护屏蔽示例由以下代码表示,VSS 表示防护网,CLK 表示屏蔽网。

```
<guard_shield port_name="CLK" shieldnet_port_name="VSS" />
```

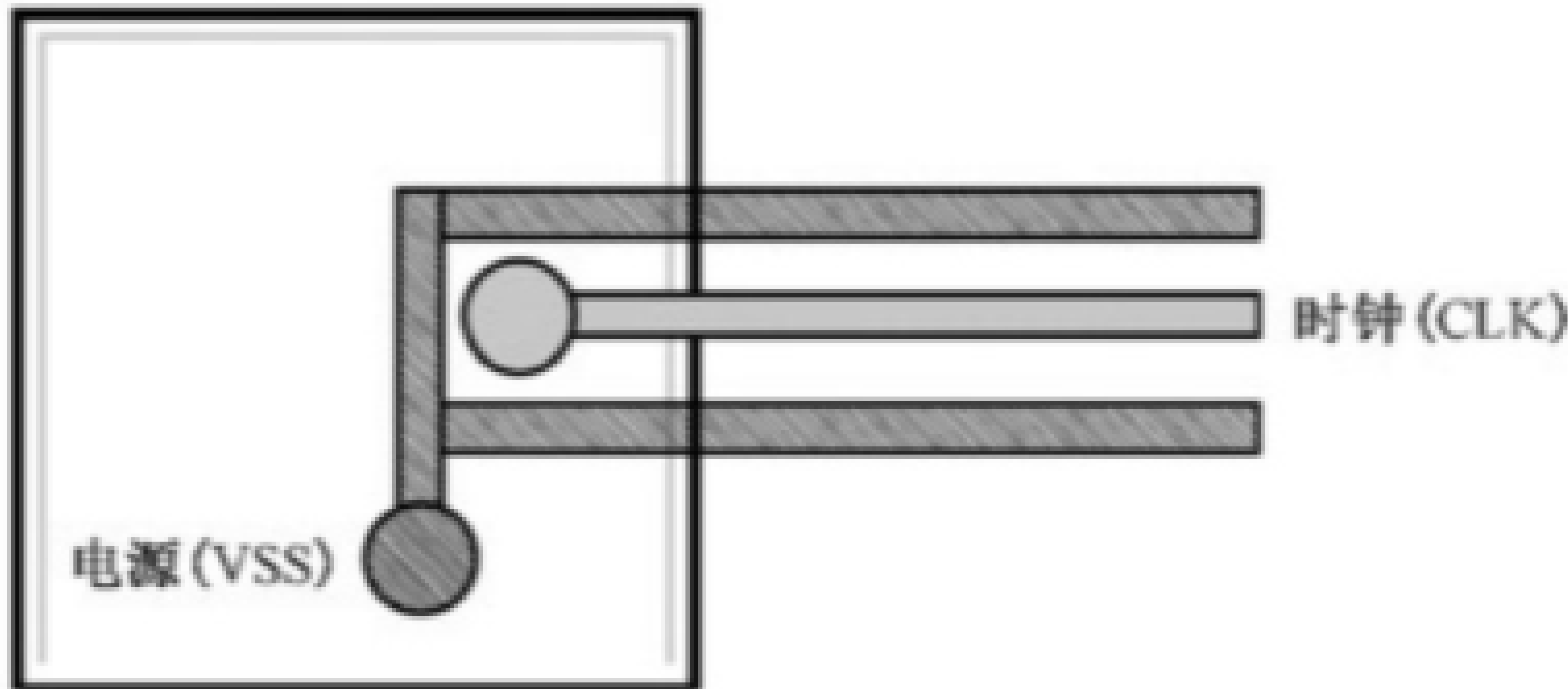


图 30 时钟信号的保护屏蔽示例

8.2.6 guideline 语句

8.2.6.1 概述

<guideline>语句定义了<module>内部布线设计的准则。<net>语句是在<netgroup>语句下,它具有 name 属性。<net name>对应于 N 格式的线路名称,该名称定义了模块内部的网络。

```
<guideline>
    [<netgroup> element]...
    [<impedance> element]...
    [<delay> element]...
    [<skew> element]...
    [<width> element]...
    [<length> element]...
    [<clearance> element]...
    [<gap> element]...
```

```
        [⟨enclosure⟩ element]...
        [⟨keepout⟩ element]...
        [⟨extension⟩ element]...
    </guideline>
```

8.2.6.2 语句组成

⟨guideline⟩语句包含以下内容：

```
⟨netgroup⟩
⟨impedance⟩
⟨delay⟩
⟨skew⟩
⟨guard shield⟩
⟨width⟩
⟨length⟩
⟨clearance⟩
⟨gap⟩
⟨enclosure⟩
⟨keepout⟩
⟨extension⟩
```

8.2.6.3 netgroup 语句

8.2.6.3.1 概述

⟨netgroup⟩语句定义了一组 BUS 信号网络。它还用于定义差分信号和相同电压组的连接。

```
⟨netgroup
    name="net_group_name"
>
    [⟨mustjoin/⟩]
    [⟨differential/⟩]
    [⟨ref_net⟩ element]...
    [⟨extensions⟩ element]...
⟨netgroup>
```

8.2.6.3.2 属性定义

⟨netgroup⟩语句的定义如下：

Name:此属性指定用于引用其他属性和语句的网络组的名称。网络组名称在单个 C 格式文件中是唯一的。

8.2.6.3.3 语句组成

⟨netgroup⟩语句包含以下内容：

```
<mustjoin>
<differential>
<ref_net>
<ref_netgroup>
<extensions>
```

8.2.6.3.4 mustjoin 语句

当指定<mustjoin>语句时,组中的网络应在模块中连接在一起。

```
<mustjoin/>
```

8.2.6.3.5 differential 语句

指定的<differential>语句表示组成差分信号的组中的网络。

```
<differential/>
```

8.2.6.3.6 ref\_net 语句

8.2.6.3.6.1 概述

<ref\_net>语句指定组成<netgroup>语句的网络。<ref\_net>对应于 N 格式的线路。

```
<ref_net
    name=="net_name"
    [polarity="polarity"]
/>
```

<ref\_net>语句可从一个或多个组中引用。

8.2.6.3.6.2 属性定义

<ref\_net>语句的属性定义如下。

- a) name:此属性指定用于从其他属性和语句引用网络的名称。该名称与 N 格式模块中的线路名称相对应。网络名称在单个 C 格式文件中是唯一的。
- b) polarity:此属性指定差分信号极性,极性应为正极或负极之一。

8.2.6.3.6.3 例句

以下是<netgroup>语句的一个示例。

```
<netgroup name="CLOCK">
    <differential/>
    <ref_net name="CKN"polarity="NEGATIVE"/>
    <ref_net name="CKQ"polarity="POSITIVE"/>
</netgroup>
<netgroup name="ADDRESS">
```

```
    <ref_net name="RADDR0"/>
    <ref_net name="RADDR1"/>
    <ref_net name="RADDR2"/>
    <ref_net name="RADDR3"/>
    <ref_net name="RADDR4"/>
    <ref_net name="RADDR5"/>
    <ref_net name="RADDR6"/>
    <ref_net name="RADDR7"/>
  </netgroup>
```

```
<netgroup name="GND">
  <mustjoin/>
  <ref_net name="GND1"/>
  <ref_net name="GND2"/>
</netgroup>
```

8.2.6.3.7 ref\_netgroup 语句

8.2.6.3.7.1 概述

<ref\_netgroup>语句用于描述<netgroup>语句的嵌套结构。

```
<ref_netgroup name="name_of_referenced_netgroup"/>
```

8.2.6.3.7.2 属性定义

<ref\_net>语句的属性定义如下：

name:此属性指定用于从其他属性和语句引用网络组的名称。网络组名称在单个 C 格式文件中是唯一的。

8.2.6.3.7.3 例句

以下是<ref\_net>语句的一个示例。

```
<netgroup name="ADDRESS_G0">
  <ref_net name="RADDR0"/>
  <ref_net name="RADDR1"/>
  <ref_net name="RADDR2"/>
  <ref_net name="RADDR3"/>
</netgroup>

<netgroup name="ADDRESS_G1">
  <ref_net name="RADDR4"/>
  <ref_net name="RADDR5"/>
  <ref_net name="RADDR6"/>
  <ref_net name="RADDR7"/>
</netgroup>

<netgroup name="ADDRESS">
```



```
    <ref_net name="ADDRESS_G0"/>
    <ref_net name="ADDRESS_G1"/>
</netgroup>
```

8.2.6.3.8 impedance 语句

8.2.6.3.8.1 通则

<impedance>语句定义了要求阻抗适配的网络的最小、典型和最大特性阻抗值。

```
<impedance
    {net_name="name_of_referenced_net"|
    group_name="name_of_referenced_net_group"}
    [type="impedance_type"]
    [min="minimun impedance"]
    typ="typical_impedance"
    [max="maximun impedance"]
/>
```

应使用网络名称(net\_name)或组名称(group\_name)属性指定网络。

8.2.6.3.8.2 属性定义

<impedance>语句的属性定义如下。

- a) net\_name:此属性指定要作为阻抗匹配目标的网络的名称,网络在 N 格式文件中定义。
- b) group\_name: 此属性指定要作为阻抗匹配目标的网络组名称。group\_name 对应于 <netgroup>语句的 name 属性。
- c) type:该属性指定特性阻抗的类型,值应为以下内容:  
single : 单端信号;  
differential : 差分阻抗;  
common : 共模阻抗。  
如果未指定类型属性,则设置 single 为默认值。
- d) typ:该属性指定典型的阻抗值,可接受的阻抗由最小和最大属性指定。阻抗的单位由<unit>语句中的<impedance>语句定义。
- e) min/max:这些属性指定可接受的阻抗值,最大和最小属性分别是最大和最小阻抗值。阻抗的单位由<unit>语句中的<impedance>语句定义。

8.2.6.3.8.3 例句

以下是<impedance>语句的一个示例。

```
<guideline>
    <netgroup name="CLOCK">
        <differential/>
        <ref_net name="CKN" polarity="NEGATIVE"/>
        <ref_net name="CKP" polarity="POSITIVE"/>
    </netgroup>
```

```
<impedance net_name="DQ1" type="single" min="45" typ="50" max="55"/>
<impedance net_name="RESET" type="single" typ="50" />
<impedance group_name="CLOCK" type="differential" typ="100" />
</guideline>
```

8.2.6.3.9 delay 语句

8.2.6.3.9.1 通则

<delay>语句定义了请求定时约束的网络的最小、典型和最大延迟值。

```
<delay
    {net_name="name_of_reference_net "|
    group_name="name_of_reference_net_group "}
    [min="minimun_delay "]
    typ="typical_delay "
    [max="maximum_delay "]
/>
```

网络应指定网络名称(net\_name)或组名称(group\_name)属性。

8.2.6.3.9.2 属性定义

<delay>语句的属性定义如下。

- a) net\_name:此属性指定要延迟约束的网络的名称,网络在 N 格式文件中定义。
- b) group\_name:此属性指定要延迟约束的网络组名称。group\_name 与<netgroup>语句的 name 属性相对应。
- c) typ:该属性指定典型的延迟值,可接受的延迟时间由最小和最大属性指定,延迟单位由<unit>语句中的<time>语句定义。
- d) min/max:这些属性指定可接受的延迟时间,最大和最小属性分别是最大和最小延迟值,延迟单位由<unit>语句中的<time>语句定义。

8.2.6.3.9.3 例句

以下是<delay>语句的一个示例。

```
<guideline>
    <netgroup name="BUS">
        <ref_net name="SIG1"/>
        <ref_net name="SIG2"/>
        <ref_net name="SIG3"/>
    </netgroup>
    <delay net_name="CLK" min="10" typ="12" max="15"/>
    <delay net_name="D3" typ="5"
    <delay netgroup="BUS" min="50" typ="55" max="60"/>
</guideline>
```

8.2.6.3.10 skew 语句

8.2.6.3.10.1 概述

⟨skew⟩语句定义了从指定网络输入和输出的信号的偏移约束。

```
⟨skew
    {net_name="name_of_net "|
    group_name="name_of_net_group "}
    [reference_net_name="name_of_net_to_be_base_delay "]
    [min="minimun_time "]
    [max="maximum_time "]
/⟩
```

网络应指定网络名称(net\_name)属性或组名称(group\_name)属性。

当指定参考网络时,偏移约束由最大时间(max)和最小时间(min)定义。在这种情况下,指定的最大时间和最小时间基于参考网络(reference\_net\_name)的传输时间。具有偏移约束的网络的传输时间要求在最小时时间和最大时间之间(参见图 31)。

```
⟨netgroup name="skewset1"⟩
    ⟨ref_net name="net1"/⟩
    ⟨ref_net name="net2"/⟩
    ⟨ref_net name="net3"/⟩
⟨/netgroup⟩
⟨skew group_name="skewset1" reference_net_name="net0"
    min="-5" max="10"/⟩
```

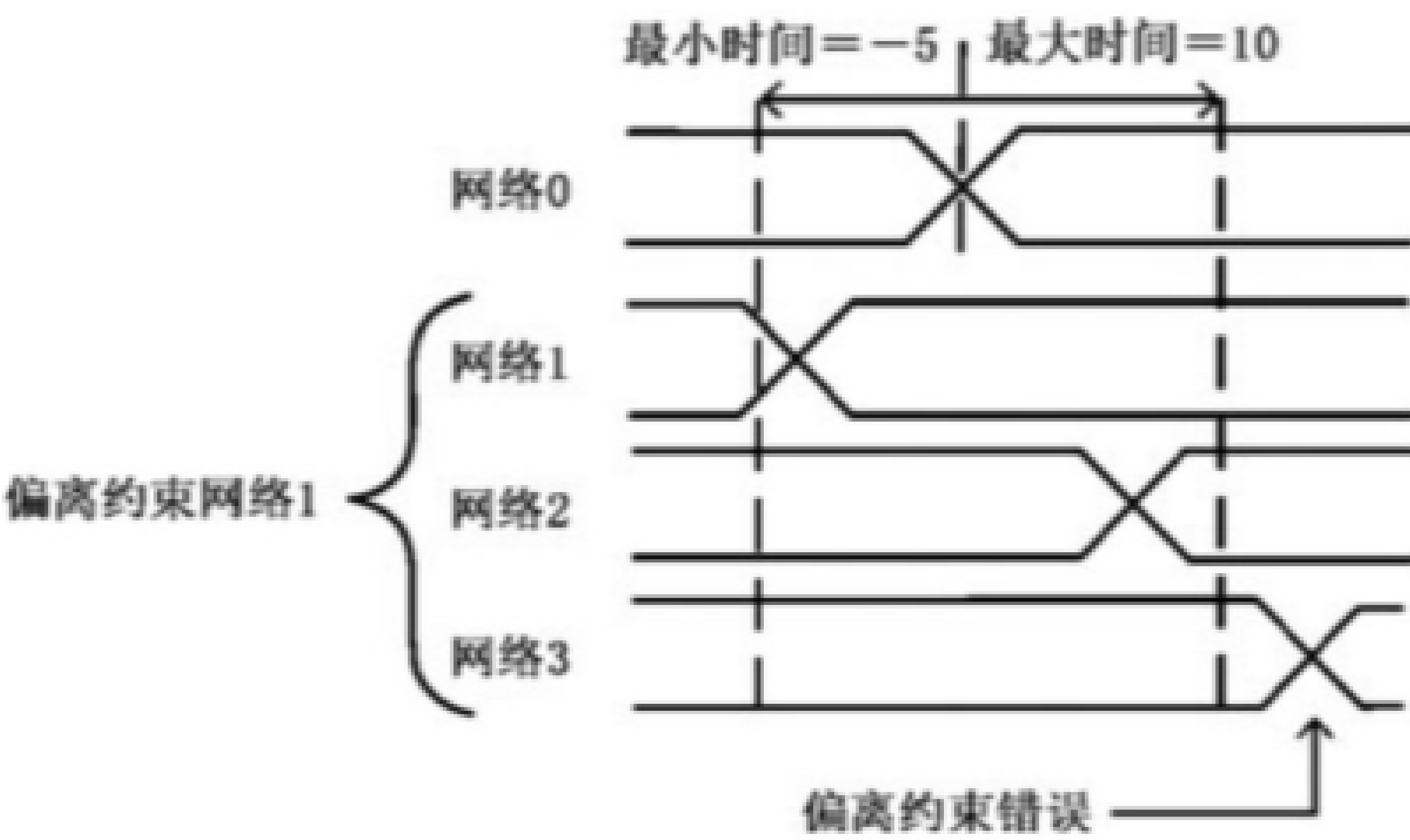


图 31 具有 min 和 max 属性的偏移约束示例

当未指定参考网络时,偏移约束由最大时间(max)或最小时间(min)定义。如果定义了最大时间,则使用传输时间最快的信号作为参考网络(参见图 32)。

```
⟨netgroup name="skewset2"⟩
    ⟨ref_net name="net4"/⟩
    ⟨ref_net name="net5"/⟩
    ⟨ref_net name="net6"/⟩
```

```
</netgroup>
<skew group_name="skewset2" max="10/"
```

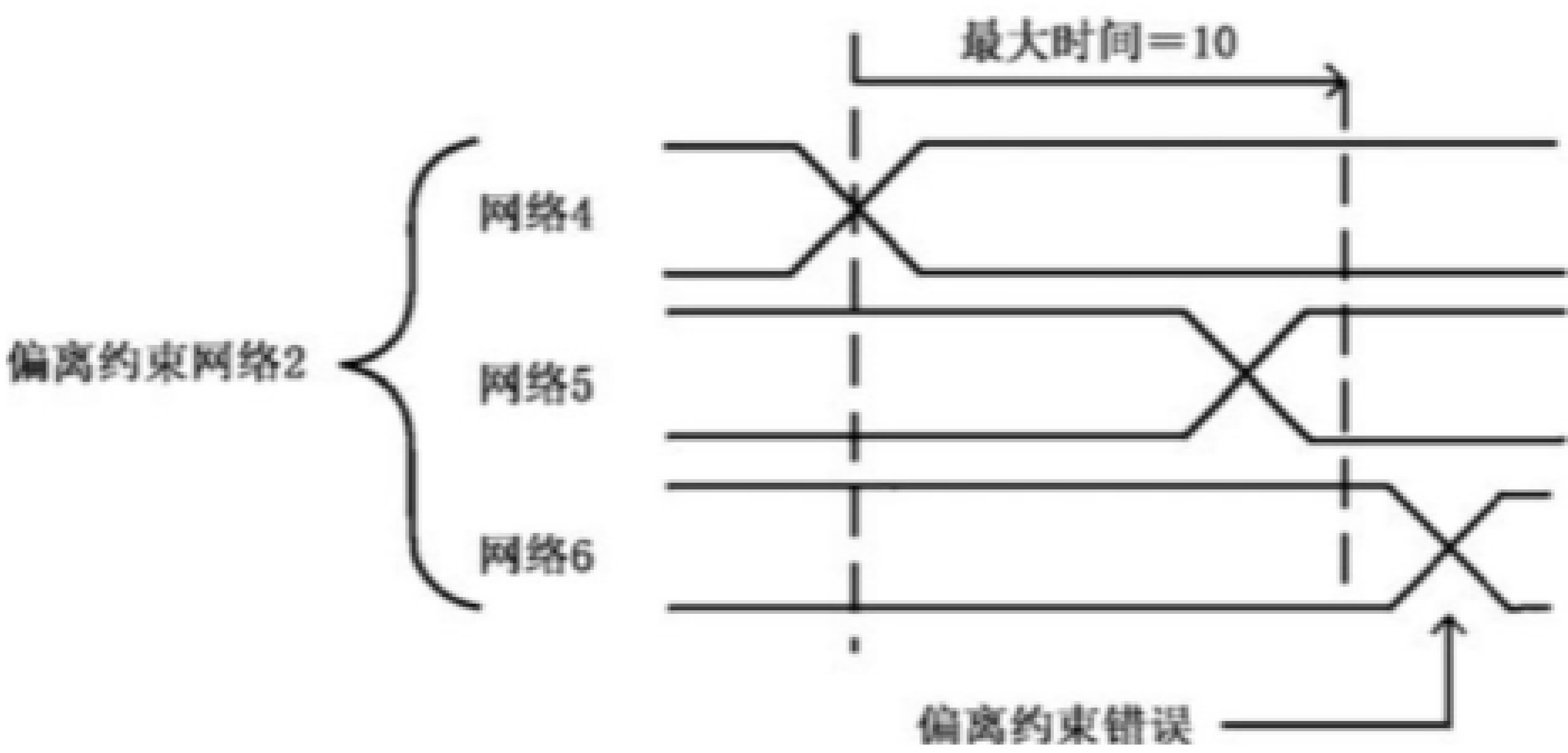


图 32 仅具有 max 属性的偏移约束示例

如果只定义了最小时间(min),则使用传输时间最慢的网络作为参考网络(参见图 33)。

```
<netgroup name="skewset3">
  <ref_net name="net7"/>
  <ref_net name="net8"/>
  <ref_net name="net9"/>
</netgroup>
<skew group_name="skewset3" min="-10/">
```

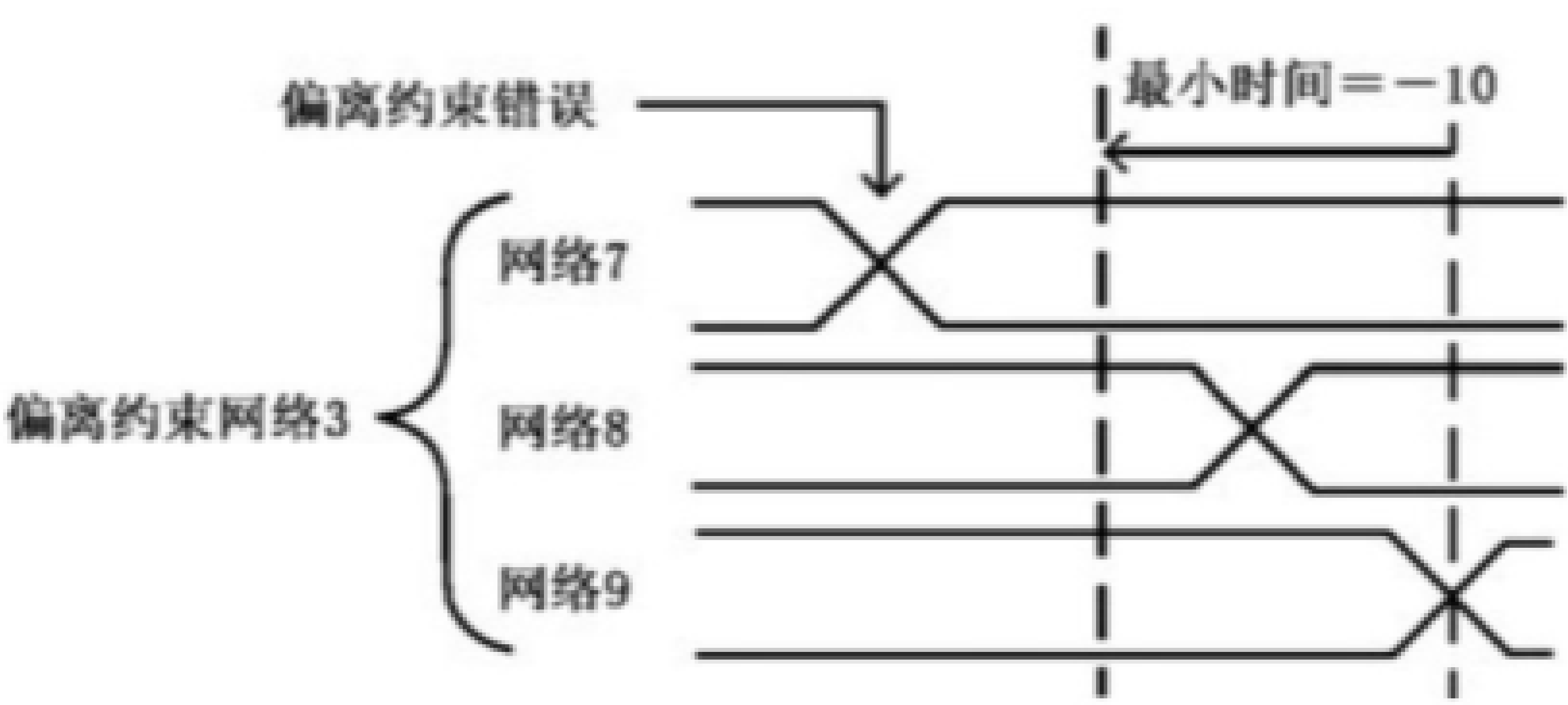


图 33 仅具有 min 属性的偏移约束示例

8.2.6.3.10.2 属性定义

- <skew>语句的属性定义如下。
- a) net\_name:此属性指定要作为<skew>语句约束目标的网络名称。指定 net\_name 属性需使用 reference\_net\_name 来定义基本延迟。
  - b) group\_name:此属性指定要作为偏斜约束目标的网络组名称。属于该组的所有网络都应处于偏移约束内。
  - c) reference\_net\_name:此属性指定要用作偏移约束的延迟的参考网络。
  - d) min/max:最大和最小属性分别指定最大和最小偏移。最大属性设置正值,最小属性设置负值。当 reference\_net\_name 属性指定参考网络时,最小和最大属性应一起使用。当未指定参考网络时,应使用最小或最大属性之一。最小和最大偏移的单位由<unit>语句中的<time>语句定义。

8.2.6.3.10.3 例句

以下是<skew>语句的一个示例。

```

<netgroup name="skewset1">
  <ref_net="net1" />
  <ref_net="net2" />
  <ref_net="net3" />
</netgroup>
<netgroup name="skewset2">
  <ref_net="net4" />
  <ref_net="net5" />
  <ref_net="net6" />
</netgroup>
<netgroup name="skewset3">
  <ref_net="net7" />
  <ref_net="net8" />
  <ref_net="net9" />
</netgroup>

<skew group_name="skewset1" reference_net_name="net0" min="-5" max="10" />
<skew group_name="skewset2" max="10" />
<skew group_name="skewset2" min="-10" />

<netgroup name="CK">
  <defferential/>
  <ref_net="CKP" polarity="POSITIVE"/>
  <ref_net="CKN" polarity="NEGATIVE"/>
</netgroup>
<skew group_net="CK" max="10" />
<skew group_net="CK" reference_net_name="CKP" min=max="-5" max="15" />

```

8.2.6.3.11 guard\_shield 语句

8.2.6.3.11.1 概述

<guard\_shield>语句定义了用于屏蔽的信号,以及需要屏蔽的信号。

```

<guard_shield
  { net_name="name_of_net_requiring_shield " |
    group_name="name_of_net_group_requiring_shield " }
  { shieldnet_net_name="name_of_net_using_shield " |
    shieldnet_net_group_name="name_of_net_group_using_shield " }
/>

```

例如,在模拟-数字混合电路设计的情况下,最好通过模拟接地来屏蔽模拟信号。<guard\_shield>语句用于定义屏蔽网的组合。

8.2.6.3.11.2 属性定义

- 〈guard\_shield〉语句的属性定义如下。
- a) net\_name:此属性指定需要屏蔽的网络名称。网络名称对应于 N 格式的模块导线名称。
  - b) group\_name:此属性指定需要屏蔽的网络组名称。属于网络组的所有网络都需要屏蔽。group\_name 与〈netgroup〉语句定义相对应。应使用 net\_name 或 group\_name 指定屏蔽目标。
  - c) shieldnet\_net\_name:该属性指定用于形成屏蔽图案的信号名称。shieldnet\_net\_name 与 N 格式的模块线路名称相对应。
  - d) shieldnet\_net\_group\_name:此属性指定用于形成屏蔽图案的网络组名称。shieldnet\_net\_group\_name 与〈netgroup〉语句相对应。shieldnet\_net\_name 或 shieldnet\_net\_group\_name 用于指定形成屏蔽信号的名称。

8.2.6.3.11.3 例句

图 34 中的保护屏蔽示例由以下代码表示,VSS 表示防护网,CLK 表示屏蔽网。

```
〈guard_shield net_name="CLK" shield_net_name="VSS" /〉
```

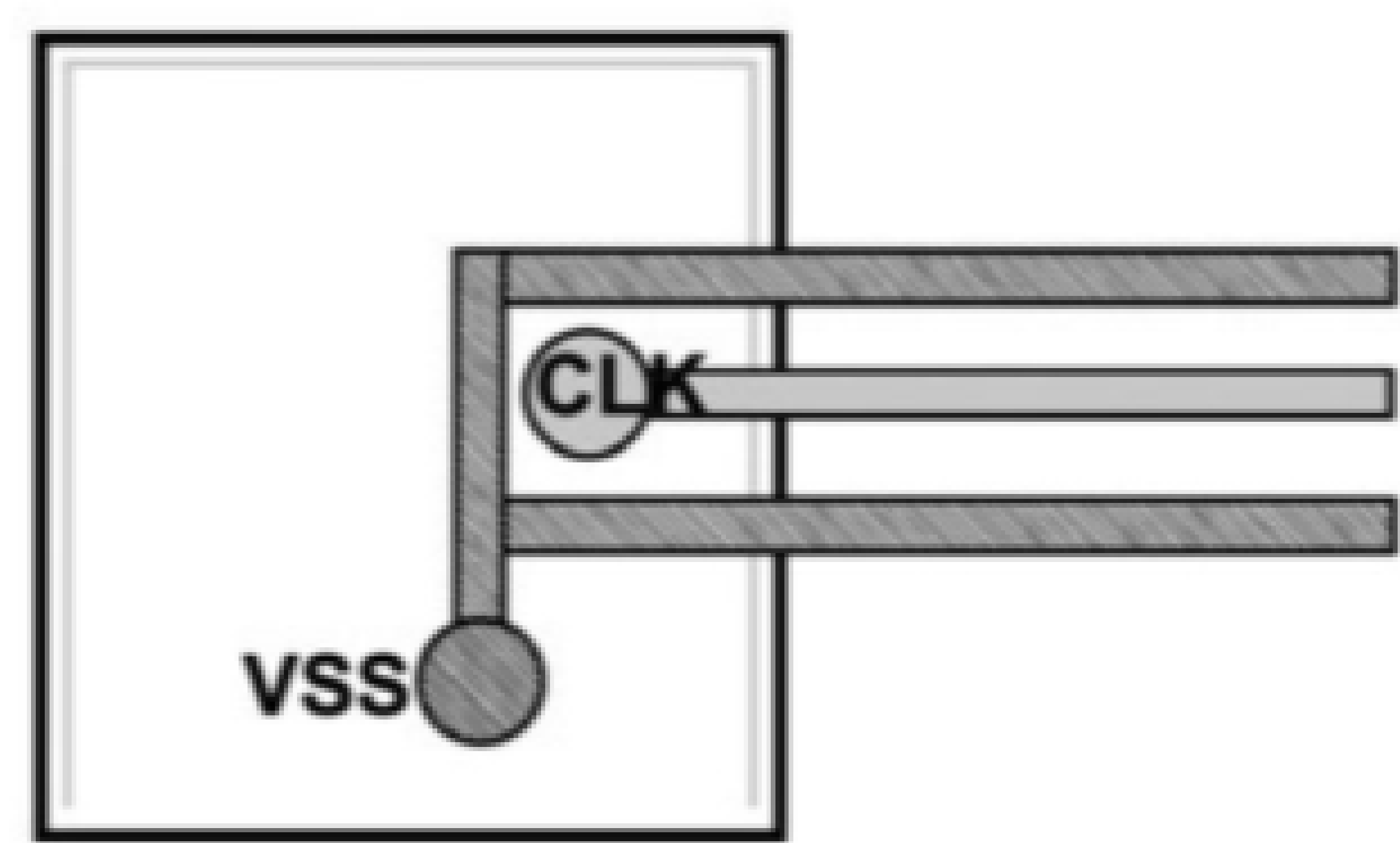


图 34 时钟信号的保护屏蔽示例

8.2.6.3.12 width 语句

8.2.6.3.12.1 概述

〈width〉语句定义用于布线的线宽。

```
〈width  
    {net name="net_name" |  
      group_name="net_group_name"}  
    [min="minimum_width"]  
    [max="miximum_width"]  
    [layer="layer_name"]  
/〉
```

此语句能指定对〈net〉语句或〈netgroup〉语句的布线宽度的约束。宽度约束还与层相关。

8.2.6.3.12.2 属性定义

〈width〉语句的属性定义如下。

- a) net\_name:此属性指定应用约束的网络名称。网络名称与 N 格式的模块线路名称相对应。
- b) group\_name:此属性指定应用约束的网络组名称。group\_name 与〈netgroup〉定义相对应。应使用 net\_name 或 group\_name 来指定应用约束的目标。
- c) min/max:最大和最小属性分别指定最大和最小宽度。最小和最大宽度的单位由〈unit〉语句中的〈distance〉语句定义。
- d) layer:此属性指定要应用宽度约束的布线层。当未指定层时,应将约束应用于所有布线层。

8.2.6.3.12.3  例句

以下是〈width〉语句的一个示例。

```
<width net_name="net1" min="100" layer="L1"/>
<width net_name="net1" min="110" layer="L2"/>
<width net_name="net1" min="110" layer="L3"/>
<width net_name="net1" min="100" layer="L4"/>
```

8.2.6.3.13  length 语句

8.2.6.3.13.1  概述

此语句指定网络或网络组布线的绝对长度,也指定约束的相关图层。

```
<length
    {net name="net_name " |
      group_name="net_group_name "}
    [min="minimum_length "]
    [max="miximum_length "]
    [layer="layer_name "]
/>
```

8.2.6.3.13.2  属性定义

〈length〉语句的属性定义如下。

- a) net\_name:此属性指定应用约束的网络名称。网络名称与 N 格式的模块线路名称相对应。
- b) group\_name:此属性指定应用约束的网络组名称。group\_name 与〈netgroup〉相对应。应使用 net\_name 或 group\_name 来指定将应用约束的目标。
- c) min/max:最大和最小属性分别指定最大和最小长度。最小和最大长度的单位由〈unit〉语句中的〈distance〉语句定义。
- d) layer:该属性指定应用长度约束的布线层。如果未指定图层,则长度约束表示总长度,而与布线图层无关。

8.2.6.3.13.3  例句

以下是〈length〉语句的一个示例。

```
<netgroup name="group1">
    <ref_net name="net4" />
    <ref_net name="net5" />
</netgroup>
<length group_name="group1" min="30000" max="30300" layer="L2"/>
```

8.2.6.3.14 clearance 语句

8.2.6.3.14.1 概述

该语句指定网络或网络组布线的间距约束,即不同网络或不同网络组之间的间距(参见图 35)。该语句能指定约束的相关图层。

```
<clearance
    {net name="net_name " |
      group_name="net_group_name "}
    [min="minimum_clearance "]
    [max="miximum_clearance "]
    [layer="layer_name "]
/>>
```

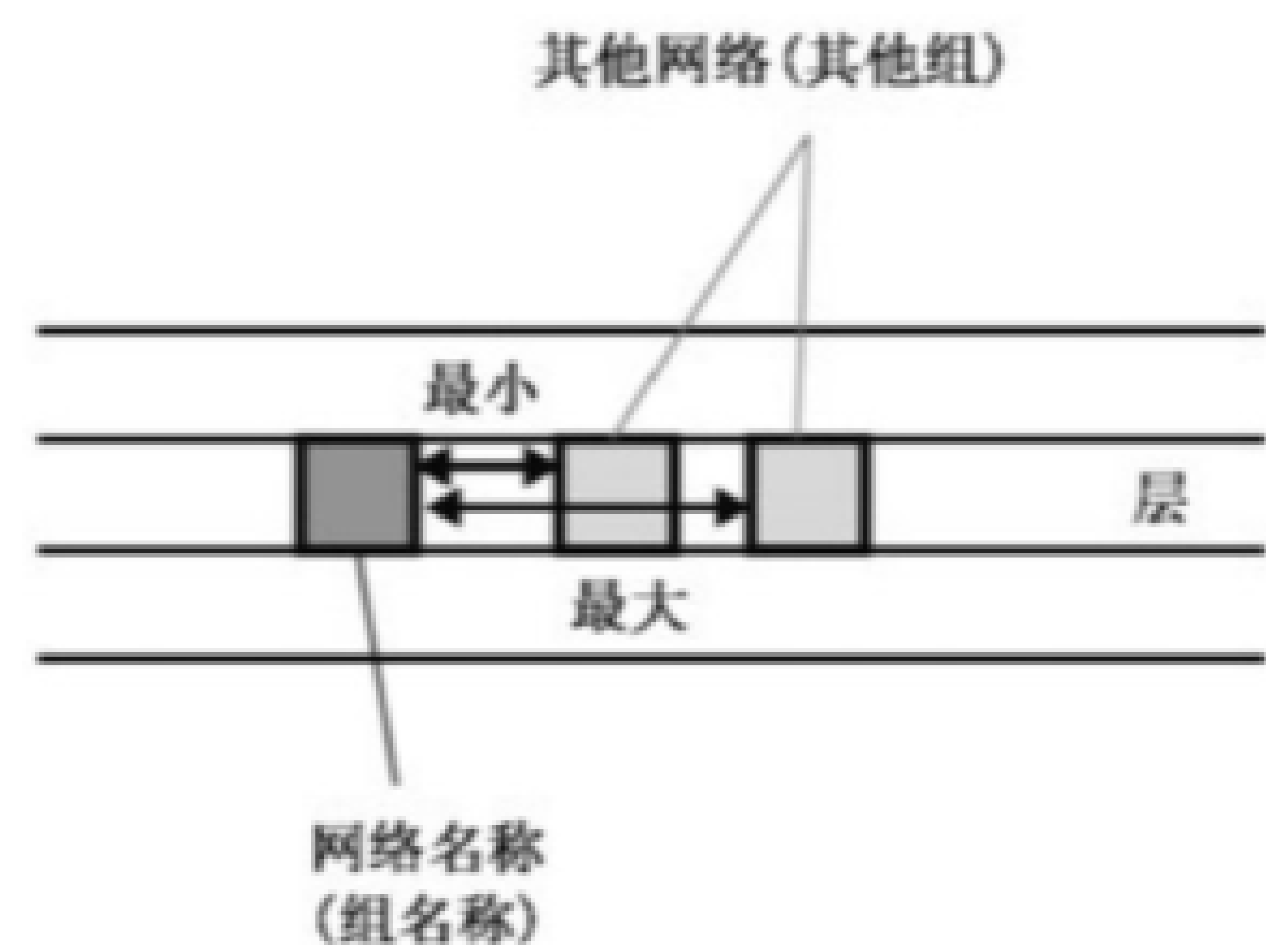


图 35 间距约束示例

8.2.6.3.14.2 属性定义

- <clearance>语句的属性定义如下。
- a) net\_name:此属性指定应用约束的网络名称。网络名称与 N 格式的模块磁安路名称相对应。
  - b) group\_name:此属性指定应用约束的网络组名称。group\_name 应与<netgroup>相对应。应使用 net\_name 或 group\_name 来指定将应用约束的目标。
  - c) min/max:最大和最小属性分别指定最大和最小间距。最小和最大间距的单位由<unit>语句中的<distance>语句定义。
  - d) layer:该属性指定应用长度约束的布线层。如果未指定图层,则长度约束表示总长度,而与布线图层无关。

8.2.6.3.14.3 例句

以下示例是一个间距约束,要求网络 AA 与 L 层中的其他网络之间的间距大于 10 μm,并且与 L2 层和 L3 层中的其他网络相比宽度大于 11 μm。

```
<clearance net_name="AA" min="100" layer="L1"/>
<clearance net_name="AA" min="110" layer="L2"/>
<clearance net_name="AA" min="110" layer="L3"/>
<clearance net_name="AA" min="100" layer="L4"/>
```



8.2.6.3.15 gap 语句

8.2.6.3.15.1 概述

该语句指定网络或网络组布线的间距约束,即属于同一网络或同一网络组之间不同网络的间距(参见图 36)。该语句还能指定约束的相关图层。

```
<gap
    {net name="net_name " |
      group_name="net_group_name "}
    [min="minimum_gap "]
    [max="miximum_gap "]
    [layer="layer_name "]
/>
```

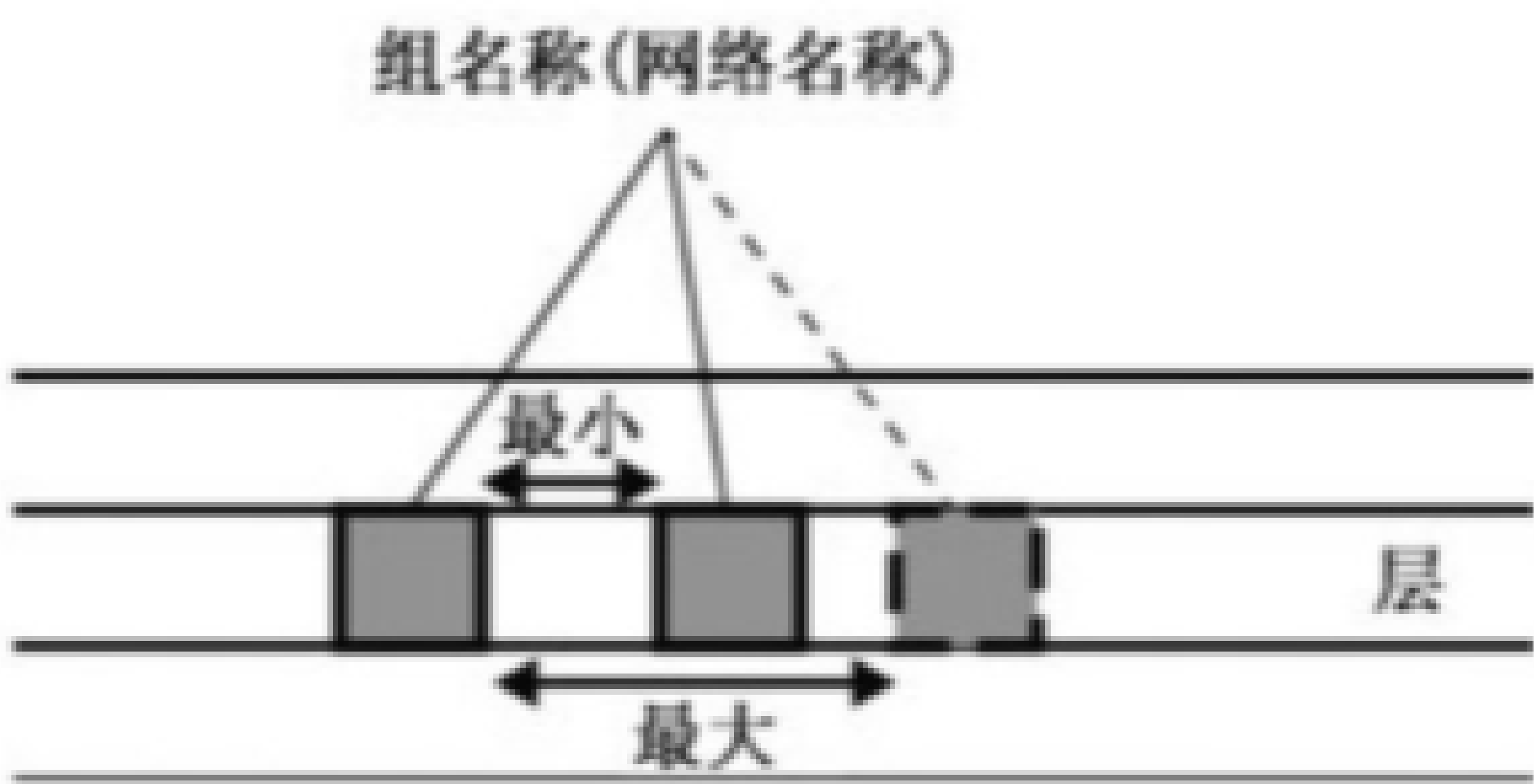


图 36 间距约束示例

<clearance>侧重于“一个网络到另一个网络”或“一个网组到另一网组”的间距,而<gap>侧重于网内或网组内间距。

8.2.6.3.15.2 属性定义

- <gap>语句的属性定义如下。
- a) net\_name:此属性指定应用约束的网络名称。网络名称与 N 格式的模块线路名称相对应。
  - b) group\_name:此属性指定应用约束的网络组名称。group\_name 应与<netgroup>相对应。使用 net\_name 或 group\_name 来指定应用约束的目标。
  - c) min/max:最大和最小属性分别指定最大和最小间距。最小和最大间距的单位由<unit>语句中的<distance>语句定义。
  - d) layer:该属性指定应用长度约束的布线层。如果未指定图层,则长度约束表示总长度,而与布线图层无关。

8.2.6.3.15.3 例句

以下示例是一个间距约束,要求属于 AA 组的网络,其布线间距为 10 μm~12 μm。

```
<gap group_name="AA" min="10" max="12" />
```

8.2.6.3.16 enclosure 语句

8.2.6.3.16.1 概述

<enclosure>语句用于指定用另一层中的金属层覆盖布线层的规则。它能够指导带状线结构或微带线结构进行布线设计。语句能指定约束的相关图层。

```
<enclosure
    {net name="net_name" |
      group_name="net_group_name "}
    [min="minimum_enclosure "]
    [layer="layer_name "]
    [upper_layer="name_of_upper_layer "]
    [lower_layer="name_of_lower_layer "]
    [upper_net="name_of_upper_net "]
    [lower_net="name_of_lower_net "]
/>
```

8.2.6.3.16.2 属性定义

<enclosure>语句的属性定义如下。

- a) net\_name:此属性指定带状线结构或微带线结构的网络名称。net\_name 对应于 N 格式的模块线路名称。
- b) group\_name:此属性指定应用约束的网络组名称。group\_name 与<netgroup>相对应。使用 net\_name 或 group\_name 来指定将应用约束的目标。
- c) min:最小属性指定金属平面覆盖特定网络导线的最小延伸长度。最小延伸的单位由<unit>语句中的<distance>语句定义。
- d) layer:该属性指定布线层,约束应仅应用于指定层上的布线金属。当未指定图层时,应将约束应用于所有布线图层。
- e) upper\_layer:当指定了 upper\_layer 属性时,布线金属的上侧应被该属性指定层上的金属平面覆盖。
- f) lower\_layer:当指定了 lower\_layer 属性时,布线金属的下侧应被该属性指定层上的金属平面覆盖。
- g) upper\_net:upper\_net 属性指定覆盖布线金属上侧的网络。如果未指定此属性,则可以使用任何网络覆盖一个网络。
- h) lower\_net:lower\_net 属性指定覆盖布线金属下侧的网络。如果未指定此属性,则可以使用任何网络覆盖一个网络。

8.2.6.3.16.3 例句

以下是<enclosure>语句的一个示例,参见图 37。

```
<enclosure group_name="net_group1" min="50"
  upper_layer="L1" lower_layer="L3"]
upper_net="SIG1" lower_net="SIG1"/>
```

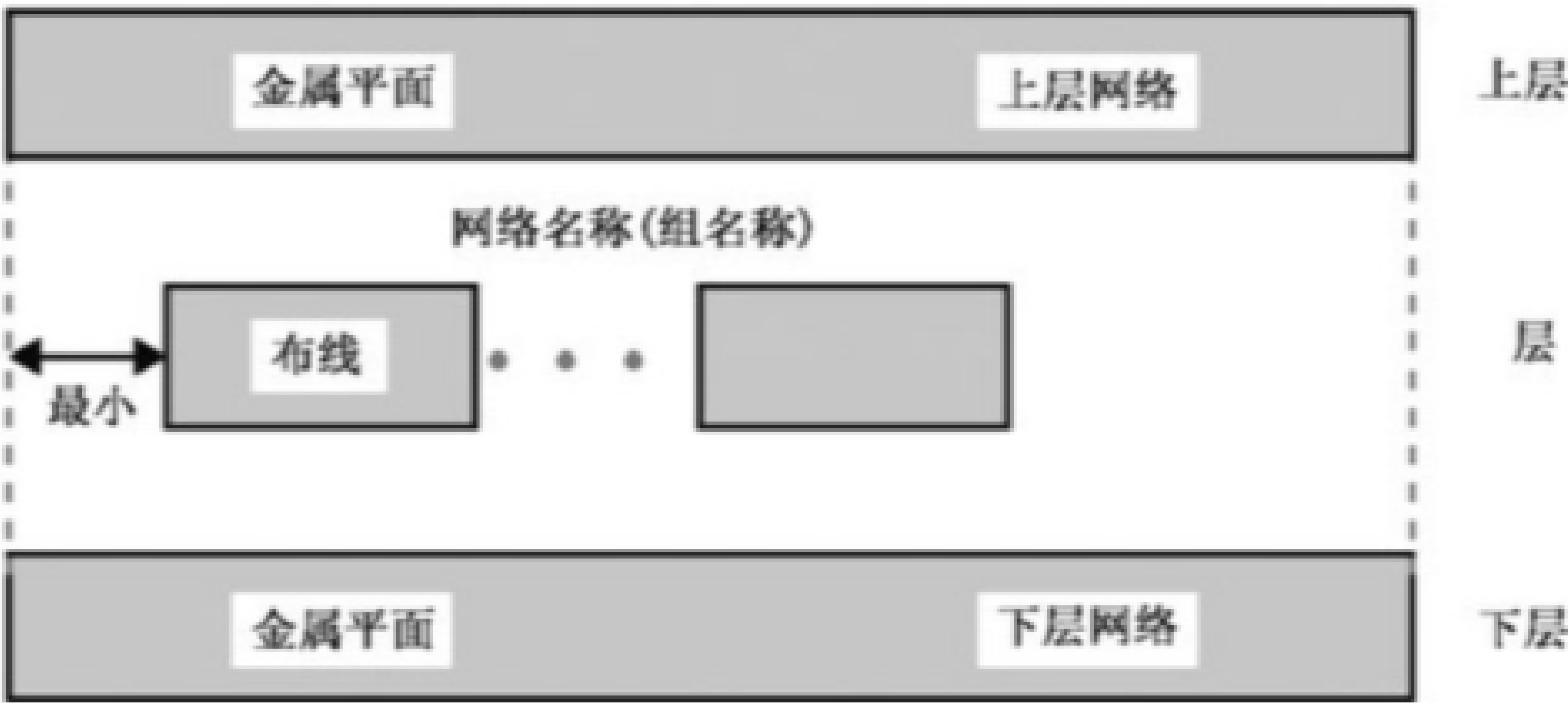


图 37 覆盖约束准则示例

8.2.6.3.17 keepout 语句

8.2.6.3.17.1 概述

该语句用于指定特定网络布线相邻层中存在的布线和金属禁止区域。该语句能指定约束的相关图层。

```
<keepout
    {net name="net_name " |
      group_name="net_group_name "}
    [min="minimum_overhang "]
    [ref_layer="layer_name "]
    [target_layer="UPPER | LOWER | ALL "]
>
    [<target_layer> element]...
    [<extensions> element]...
</keepout>
```

8.2.6.3.17.2 属性定义

<keepout>语句的属性定义如下。

- a) net\_name:该属性指定网络的名称,该网络需要在其布线图案的相邻层处设置布线和金属禁止区域。net\_name 与 N 格式的模块线路名称相对应。
- b) group\_name:此属性指定网络组的名称,该网络组需要在其布线图案的相邻层处设置布线禁止区域。group\_name 与<netgroup>语句相对应。
- c) min:最小值属性指定禁止布线区域的最小延伸量。禁止布线区域的形状是指定网络或网络组的路由区域按最小属性定义的距离的扩展。最小值的单位由<unit>语句中的<distance>语句定义。
- d) ref\_name:此属性指定由 net\_name 或 group\_name 给定的特定网络的布线层。ref\_layer 由 R 格式定义的名称指定。
- e) target\_layer: 此属性指定禁止布线和金属区域的图层。target\_layer 应是以下值之一: UPPER、LOWER 或 ALL。UPPER 指定除 ref\_layer 之外的所有相对上层。LOWER 指定除 ref\_layer 之外的所有相对较低的层。ALL 指定除 ref\_layer 指定的层之外的所有层。

8.2.6.3.17.3 例句

布线层之上设置禁步层示例参见图 38a)。

```
<keepout
  group_name="BUS1"
  min="50"
  ref_layer="L5"
  target_layer="UPPER" />
```

布线层之下设置禁步层示例参见图 38b)。

```
<keepout
  group_name="BUS1"
  min="50"
  ref_layer="L5"
  target_layer="LOWER" />
```

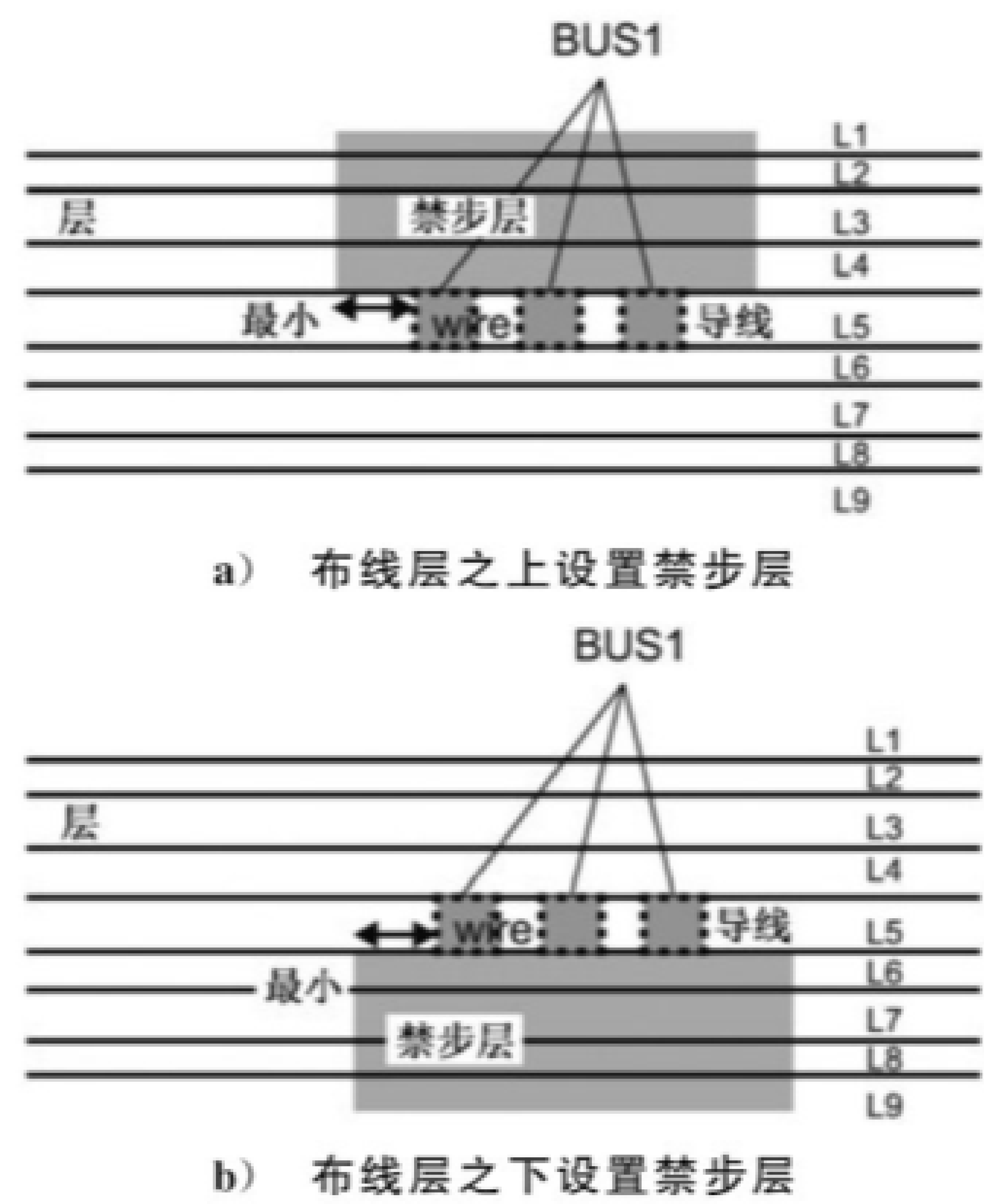


图 38 禁止布线区域示例

8.2.6.3.18 target\_layer 语句

8.2.6.3.18.1 概述

<target\_layer>语句指定布线和金属禁止层,可在一个<keepout>下多次使用。

```
<target_layer
  ref_layer="name_of_target_layer "
/>
```

8.2.6.3.18.2 属性定义

<target\_layer>语句的属性定义如下：  
ref\_layer:该属性指定禁止在其上应用布线和金属的目标图层。

8.2.6.3.18.3 例句

布线层之上设置禁步层和目标层示例参见图 39a)。

```
<keepout
  group_name="BUS1"
  min="50"
  ref_layer="L5"
  target_layer="UPPER" >
```

```
<target_layer ref_layer="L6"/>
</keepout>
```

多层目标层示例参见图 39b)。

```
<keepout
  group_name="BUS1"
  min="50"
  ref_layer="L5"
  <target_layer ref_layer="L4" />
  <target_layer ref_layer="L6" />
</keepout>
```

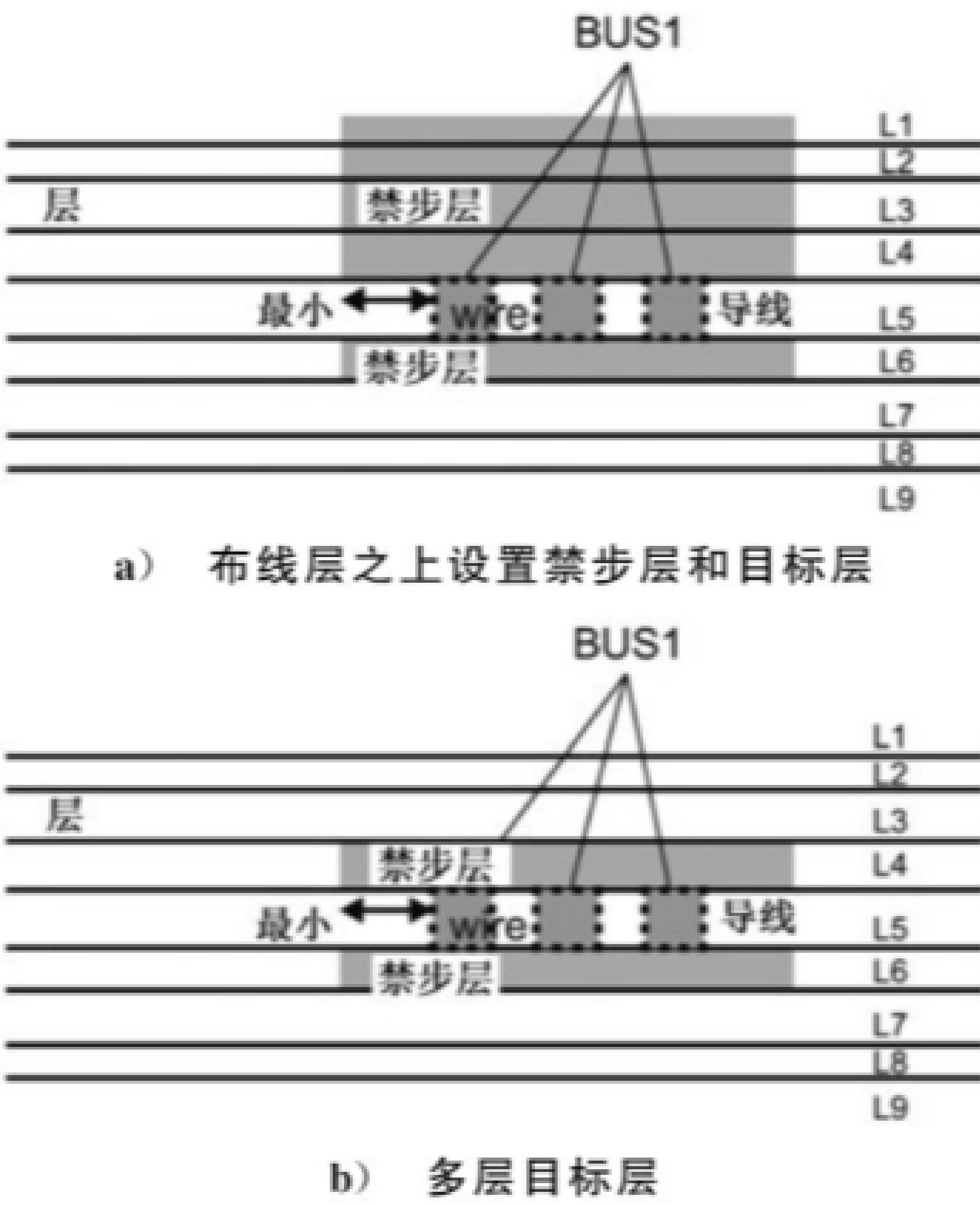


图 39 有目标层的禁止布线区域示例

8.2.7 specification 语句

8.2.7.1 概述

<specification>语句定义了模块本身的规范,例如功耗。

```
<specification>
  [<power> element ]
  [<resistance> element ]
  [<capacitance> element ]
  [<inductance> element ]
  [<extensions> element ]...
</specification>
```

8.2.7.2 语句组成

<specification>语句包含以下内容:

<power>  
 <resistance>  
 <capacitance>  
 <inductance>  
 <extensions>

8.2.7.3 power 语句

8.2.7.3.1 概述

<power>语句定义模块的功耗。

```

<power
  [min="minimum_power_consumption "]
  typ="typical_power_consumption "
  [max="maximum_power_consumption "]
/>

```

8.2.7.3.2 定义属性

<power>语句的属性定义如下：

- a) typ:此属性指定典型的功耗,功耗单元由<unit>语句中的<power>语句定义；
- b) min/max:此属性指定了功耗的扰动,最大和最小属性分别是最大和最小功耗,功率单元由<unit>语句中的<power>语句定义。

8.2.7.3.3 例句

以下是<power>语句的示例。

```

<specification>
  <power typ="1.0" />
</specification>

```

8.2.7.4 resistance 语句

8.2.7.4.1 概述

<resistance>语句指定模块的电阻。

```

<resistance
  [min="minimum_resistance "]
  typ="typical_resistance "
  [max="maximum_resistance "]
/>

```

在 <resistance>,<capacitance> 和 <inductance> 语句中定义的电阻、电容和电感是一个相关联的系列。

8.2.7.4.2 属性定义

- ⟨resistance⟩语句的属性定义如下。
- a) typ:该属性指定典型的电阻值。电阻的单位由⟨unit⟩语句中的⟨resistance⟩语句定义。
  - b) min/max:这些属性指定了电阻的扰动。最大和最小属性分别是最大电阻和最小电阻。电阻的单位由⟨unit⟩语句中的⟨resistance⟩语句定义。

8.2.7.4.3 例句

以下是⟨resistance⟩语句的示例。

```
⟨specification⟩  
    ⟨resistance typ="1.0" /⟩  
⟨/specification⟩
```

8.2.7.5 capacitance 语句

8.2.7.5.1 概述

⟨capacitance⟩语句表示模块的电容。

```
⟨capacitance  
    [min="minimum_capacitance "  
    typ="typical_capacitance "  
    [max="maximum_capacitance "  
/⟩
```

在⟨resistance⟩,⟨capacitance⟩和⟨inductance⟩语句中定义的电阻、电容和电感是一个相关联的系列。

8.2.7.5.2 属性定义

- ⟨capacitance⟩语句的属性定义如下。
- a) typ:该属性指定典型的电容值。电容的单位由⟨unit⟩语句中的⟨capacitance⟩语句定义。
  - b) min/max:这些属性指定电容的扰动。最大和最小属性分别是最大和最小电容。电容的单位由⟨unit⟩语句中的⟨capacitance⟩语句定义。

8.2.7.5.3 例句

以下是⟨capacitance⟩语句的示例。

```
⟨specification⟩  
    ⟨capacitance typ="1.0" /⟩  
⟨/specification⟩
```

8.2.7.6 inductance 语句

8.2.7.6.1 概述

⟨inductance⟩语句指定模块的电感。

```
<inductance
  [min="minimum_inductance "]
  typ="typical_inductance "
  [max="maximum_inductance "]
/>
```

在 <resistance> , <capacitance> 和 <inductance> 语句中定义的电阻、电容和电感是一个相关联的系列。

8.2.7.6.2 属性定义

- <inductance> 语句的属性定义如下。
- a) typ: 此属性指定典型电感。电感的单位由 <unit> 语句中的 <inductance> 语句定义。
  - b) min/max: 这些属性说明了电感的扰动。最大和最小属性分别是最大和最小电感。电感的单位由 <unit> 语句中的 <inductance> 语句定义。

8.2.7.6.3 例句

以下是 <inductance> 语句的示例。

```
<specification>
  <inductance typ="1.0" />
</specification>
```

8.2.8 surrounding\_characteristics 语句

8.2.8.1 概述

<surrounding\_characteristics> 语句提供了关于材料三维质量的物理特性的信息, 在这里表示为模块被放置的周围环境, 周围的形状是矩形的盒子形。

```
<surrounding_characteristics
  depth="depth "
  width="width "
  height="height "
  x="x_coordinate "
  y="y_coordinate "
  z="z_coordinate "
  [thermal_conductivity="thermal_conductivity "]
  [emissivity="emissivity "]
  [specific_heat_capacity="specific_heat_capacity "]
  [density="density "]
  [power="power_consumption "]
>

  [<extension> element]...
```



</surrounding\_characteristics>

8.2.8.2 属性定义

<surrounding\_characteristics>语句的属性定义如下,见图 40。

- a) depth:该属性指定围绕物体的深度(沿 Y 轴的大小)。深度单位由<unit>语句中的<distance>语句定义。
- b) width:该属性指定围绕物体的宽度(沿 X 轴的大小)。宽度单位由<unit>语句中的<distance>语句定义。
- c) height:该属性指定围绕物体的高度(沿 Z 轴的大小)。高度单位由<unit>语句中的<distance>语句定义。

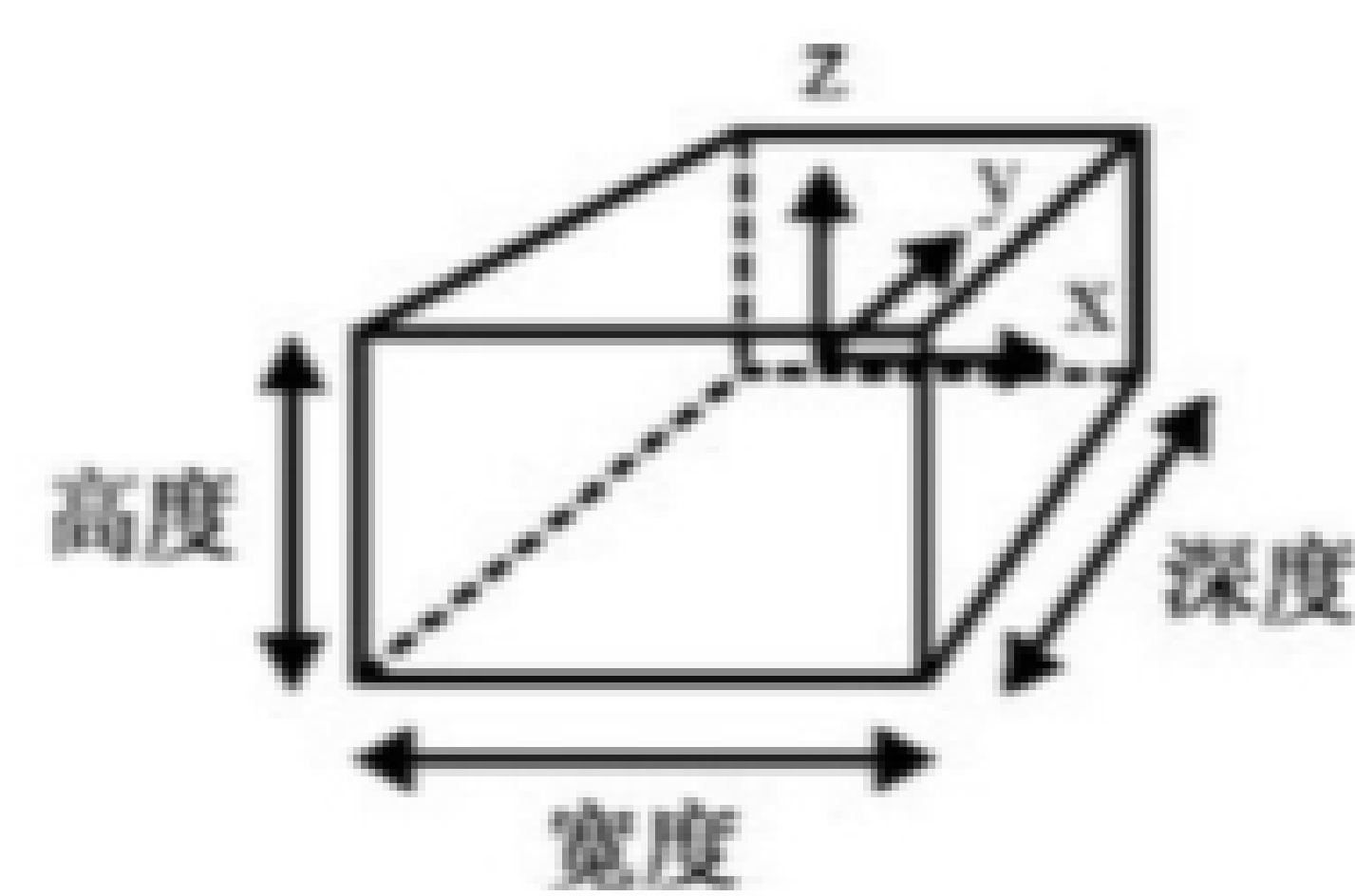


图 40 <surrounding\_characteristics>语句的深度、宽度和高度属性的示例图

- d) x、y、z: 这些属性指定围绕物体中心点的 X、Y 和 Z 坐标。坐标单位由<unit>语句中的<distance>语句定义。
- e) thermal\_conductivity:该属性指定围绕物体的热导率。导热系数的单位由<unit>语句中的<thermal\_conduction>语句定义。
- f) emissivity:该属性指定围绕物体的辐射率。
- g) specific\_heat\_capacity:该属性指定了围绕物体的比热容。specific\_heat\_capacity 的单位是由<unit>语句中的<specific\_heat\_capacity>语句定义的。
- h) density:该属性指定了围绕物体的密度。密度的单位由<unit>语句中的<density>语句定义。
- i) power:该属性指定了围绕物体所消耗的功率,不包括模块功耗。功率的单位是由<unit>语句中的<power>语句定义的。

8.2.8.3 例句

以下是<surrounding\_characteristics>语句的示例。

```
<surrounding_characteristics
  depth="150000" width="150000" height="250000"
  x="50000" y="40000" z="100000"
  thermal_conductivity="0.02"
  emissivity="0.05"
  specific_heat_capacity="1005"
  density="1.293"
  power="0.0" />
```

8.2.9 blockage 语句

8.2.9.1 概述

blockage 是一个二维区域,禁止放置或布线。该语句定义了一个被禁止区域的集合。

```
<blockage>
    [<placement> element]...
    [<routing> element]...
    [<extensions> element]...
</blockage>
```

8.2.9.2 语句内容

<blockage>语句包含以下内容：

<placement>

<routing>

<extensions>

8.2.9.3 placement 语句

8.2.9.3.1 概述

<placement>语句定义了一个区域，在该区域中，任何子模块都不应被放置在该区域的边界之外。

```
<placement
    shape_id="identifier_of_referenced_shape "
    [x="x_coordinate "]
    [y="y_coordinate "]
    [angle="angle "]
    [mount="mount_type " |
        {ref_layer="name_of_referenced_layer "
            [attach="side_to_be_attached "]}
    ]
    [stack="stack_level"]
/>
```

8.2.9.3.2 属性定义

- <placement>语句的属性定义如下。
- a) shape\_id:该属性指定预定义形状的标识符，以定义填充区域的边界形状。指定的形状应在<shape>语句中定义。
  - b) x/y:这些属性定义组件的参考点的位置。x 和 y 属性分别指定 x 坐标和 y 坐标。坐标的单位由<unit>语句中的<distance>语句定义。忽略时使用零作为默认值。
  - c) angle:该属性定义逆时针旋转相对于组件的参考点的角度，如果没有指定角度，则将零设置为默认值。旋转角度的单位由<unit>语句中的<angle>语句定义。
  - d) mount:此属性指定放置填充区域的层。子模块不应放置在指定层提供的填充区域。该属性的关键字应为以下内容之一（参见图 41），mount 属性或 ref\_layer 属性应都存在：
    - TOP:填充区域在基板的顶层；
    - BOTTOM:填充区域在基板的底层；
    - MIDDLE:填充区域在除顶、底层之外的所有内层。

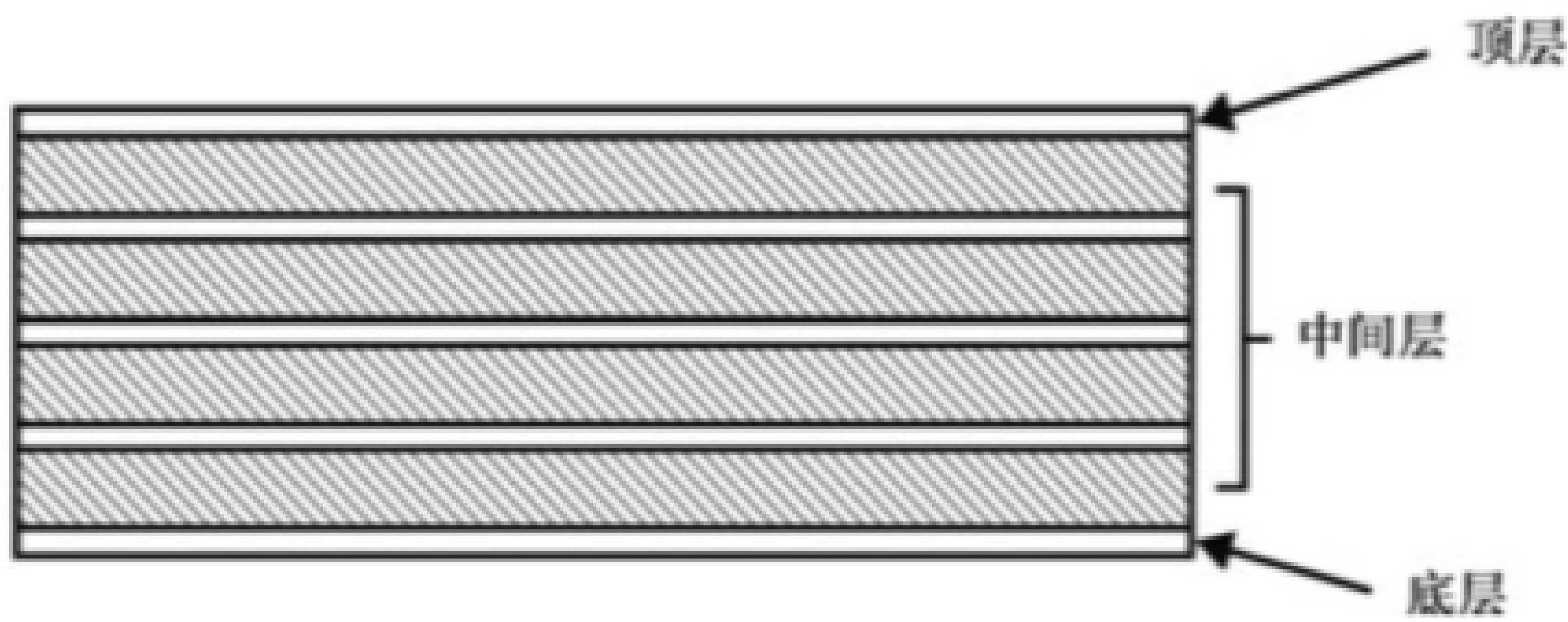


图 41 填充区域放置在安装层的说明图示

- e) `ref_layer`: 该属性通过引用在 M 格式中组合的 R 格式中定义的名称来指定基板的特定层。如果指定了该属性,则不应将子模块放置在指定层提供的填充区域的内部。`mount` 属性或 `ref_layer` 属性应都存在。
- f) `attach`: 此属性与 `ref_layer` 属性一起使用,并指定填充区域的放置边。可用的变量为 ABOVE 或 BELOW(参见图 42)。如果指定了“ABOVE”,则在指定层的上方构建放置阻塞区域。如果指定了 BELOW,则在指定的层下方构建放置填充区域。

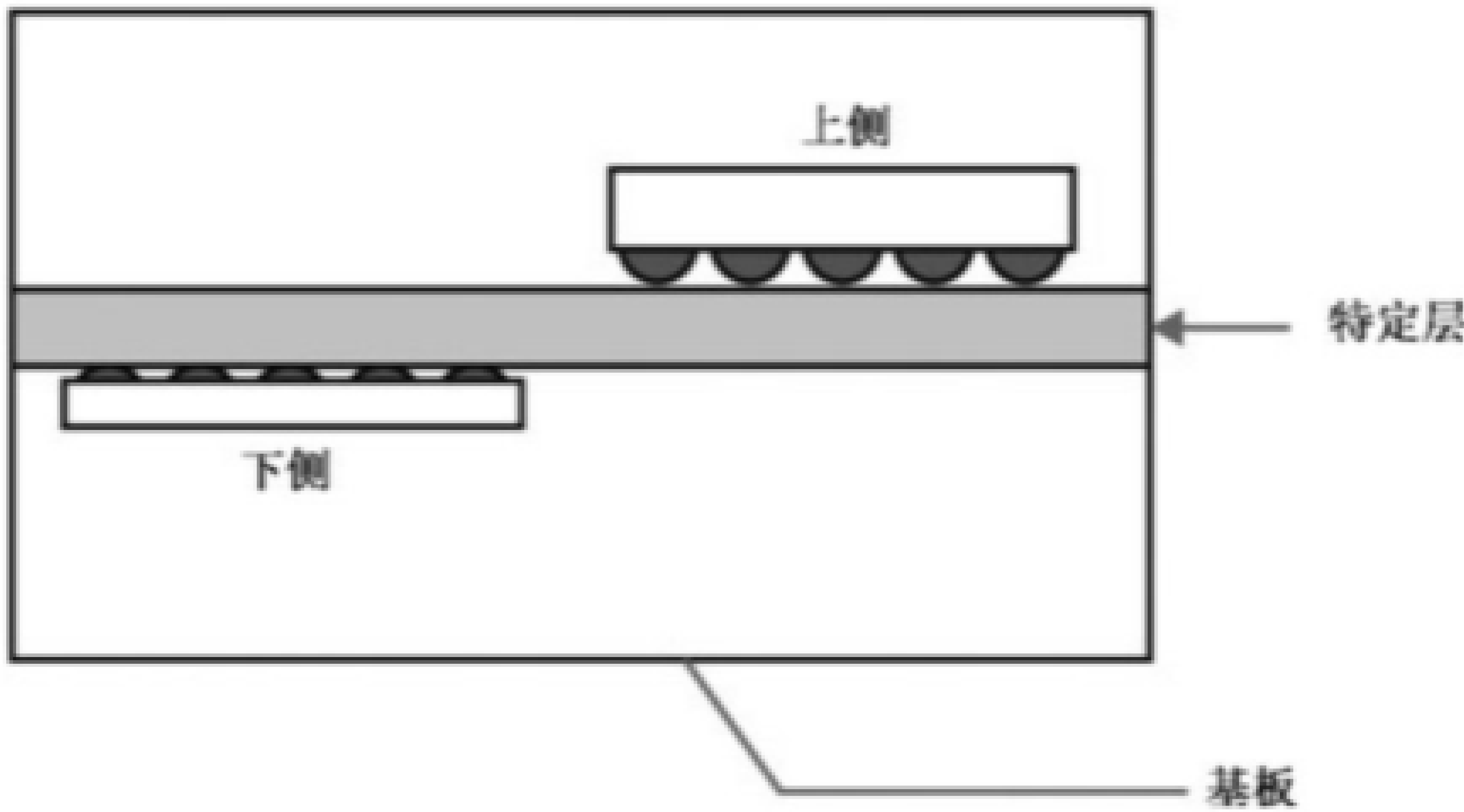


图 42 指定层安装面说明图示

- g) `stack`: 该属性指定子组件放置的“堆叠级别”(有关堆栈属性的概念,请参见图 47)。这种堵塞仅对放置在指定堆叠高度的子组件有效。如果省略此属性,则表示堆栈级别为默认级别。

8.2.9.3.3 例句

以下是<placement>语句的示例。

```
<placement shape_id="1" x="0" y="0" angle="0" mount="TOP"/>
```

8.2.9.4 routing 语句

8.2.9.4.1 概述

<routing>语句定义了一个区域,在该区域中,任何基板布线对象都不应放置在该区域边界之外。布线对象是指用于布线的金属材料,如导线、过孔和平面。

```
<routing
    Shape_id="identifier_of_referenced_shape "
    [x="x_coordinate "]
    [y="y_coordinate "]
    [angle="angle "]
```

```
[mount="mount_type"|
  ref_layer="name_of_reference_layer"]
/>>
```

8.2.9.4.2 属性定义

<routing>语句的属性定义如下。

- a) shape\_id:该属性指定预定义形状的标识符,以定义填充区域的边界形状。指定的形状应在同一文件中的<shape>语句中定义。
- b) x/y:这些属性定义组件的参考点的位置。x 和 y 属性分别指定 x 坐标和 y 坐标。坐标的单位由<unit>语句中的<distance>语句定义。忽略时使用零作为默认值。
- c) angle:该属性定义逆时针旋转相对于组件的参考点的角度,如果没有指定角度,则将零设置为默认值。旋转角度的单位由<unit>语句中的<angle>语句定义。
- d) mount:此属性指定放置填充区域的层。基板布线不应穿过指定层提供的填充区域内部。该属性的关键字为以下内容之一(参见图 43),mount 属性或 ref\_layer 属性应都存在:

- TOP:填充区域在基板的顶层;
- BOTTOM:填充区域在基板的底层;
- MIDDLE:填充区域在除顶、底层之外的所有内层。

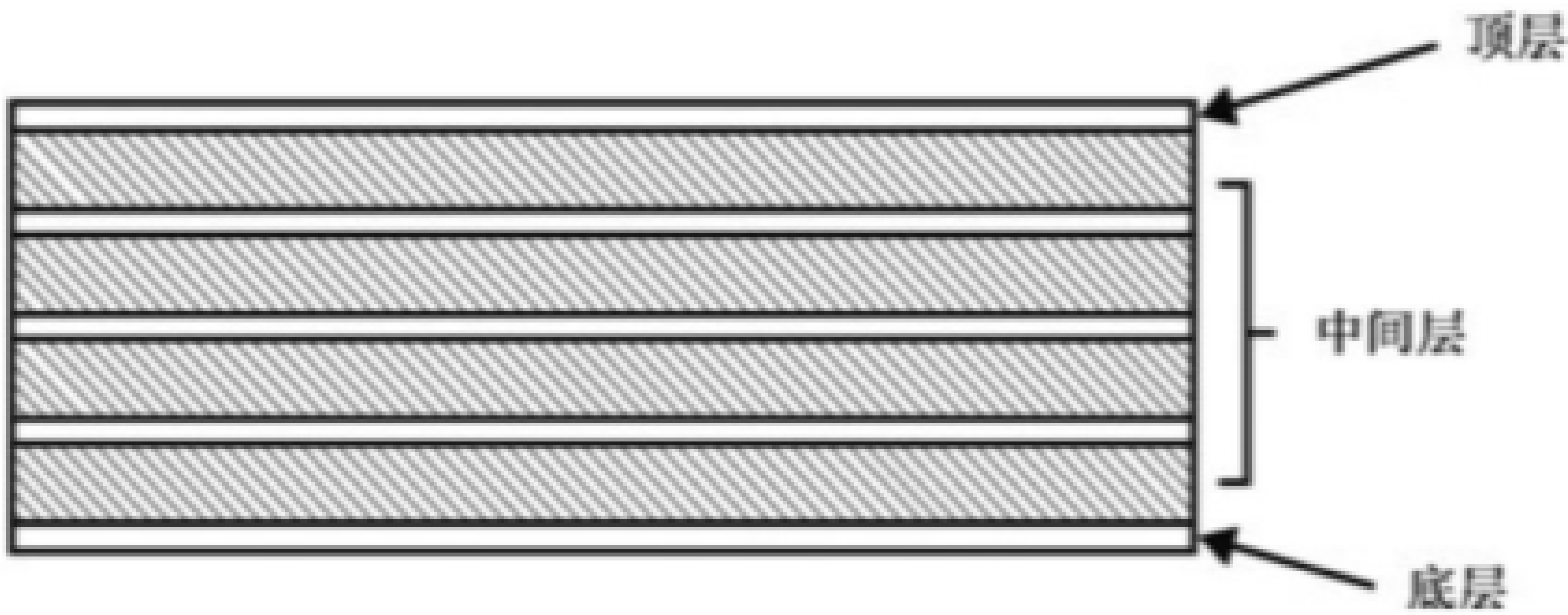


图 43 填充区域放置在安装层的说明图示

- e) ref\_layer:该属性通过引用在 M 格式中组合的 R 格式中定义的名称来指定基板的特定层。如果指定了该属性,则基板布线不应穿过指定层的填充区域内部。mount 属性或 ref\_layer 属性应都存在。

8.2.9.4.3 例句

以下是<routing>语句的示例。

```
<routing mount="TOP" shape_id="1" x="0" y="0" angle="0"/>
```

8.2.10 component 语句

8.2.10.1 概述

<component>语句实例化子模块和引线。

```
<component
  [ref_rule_name="name_of_reference_rule"]
>
  [<placement>element]...
  [<bondingwire>element]...
```

```
[<extensions>element]...  
</component>
```

8.2.10.2 属性定义

<component>语句属性定义如下：  
ref\_rule\_name:该属性通过识别合并并在 M 格式中的 R 格式中<Physicaldesign>语句中定义的名称来指定放置子模块和引线的设计规则。

8.2.10.3 语句组成

<component>语句包含以下内容：

```
<placement>  
<bondingwire>  
<extensions>
```

8.2.10.4 placement 语句

8.2.10.4.1 概述

<placement>语句定义了如何放置模块。

```
<placement  
    ref_module="name_of_referenced_module "  
    inst="instance_name "  
    [symbol="name_of_symbol "  
    [distance="unit_of_length "  
    [angleunit="unit_of_angle "  
    [scale="geometrical_scale "  
    [x="x_coordinate "  
    [y="y_coordinate "  
    [z="z_coordinate "  
    [flip="flip_type "  
    [angle="rotation_angle "  
    [mount="mount_type "|  
    ref_layer="name_of_referenced_layer "  
    [attach="layer_side_to_be_attach "  
    ]  
    [stack="order_of_loading "  
    [ref_rule_name="name_of_referenced_rule "  
    [sizing="sizing "  
/>
```

8.2.10.4.2 属性定义

<placement>语句属性定义如下。

- a) `ref_module`:此属性通过引用模块的 `name` 属性来指定要放置的模块。
- b) `inst`:此属性定义组件的示例名称,示例名称在同一个文件中是唯一的。
- c) `symbol`:此可选属性指定要放置的组件的别名。
- d) `distance`:此属性指定 `x/y` 坐标的单位。如果未指定此属性,则使用在 `<unit>` 语句中的 `<distance>` 语句处定义的单位。该值应为以下值之一:
  - `pm` 皮米;
  - `nm` 纳米;
  - `μm` 微米;
  - `mm` 毫米;
  - `m` 米。
- e) `angleunit`:此属性定义角度的单位,如果未指定此属性,则使用在 `<unit>` 语句中的 `<angle>` 语句处定义的单元,参数应为下列任何一项:
  - `degree`;
  - `radian`。
- f) `scale`:此属性定义距离的比例率,比例率应大于零,如果未指定此属性,则设置 1.0 为默认值。
- g) `x/y`:这些属性定义组件的参考点的位置,`x` 和 `y` 属性分别指定 `x` 坐标和 `y` 坐标,坐标的单位由 `<unit>` 语句中的 `<distance>` 语句定义。
- h) `z`:此属性定义组件的 `z` 坐标,即组件从顶层起始的高度,如图 44 所示,如果未指定此属性,则将零设置为默认值。

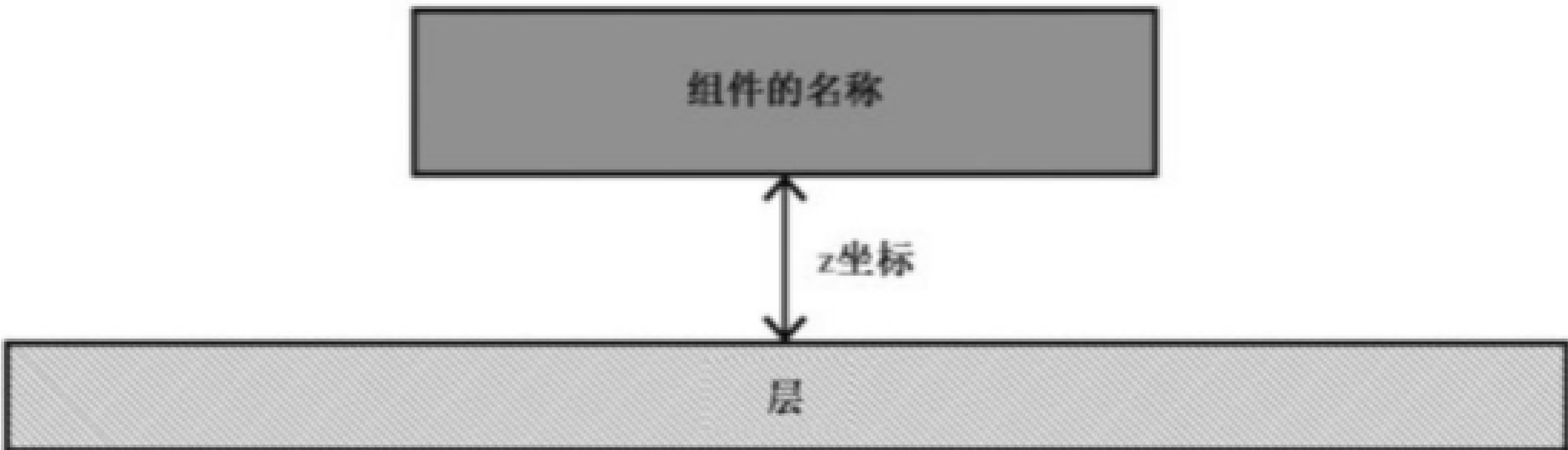


图 44 组件 `z` 坐标示意图

- i) `flip`:此属性定义倒装的类型,参数应为下列任何一项:
  - `X` 在 `X` 轴上倒装;
  - `Y` 在 `Y` 轴上倒装。

图 45 显示了 `X` 方向倒装和 `Y` 方向倒装的例子。

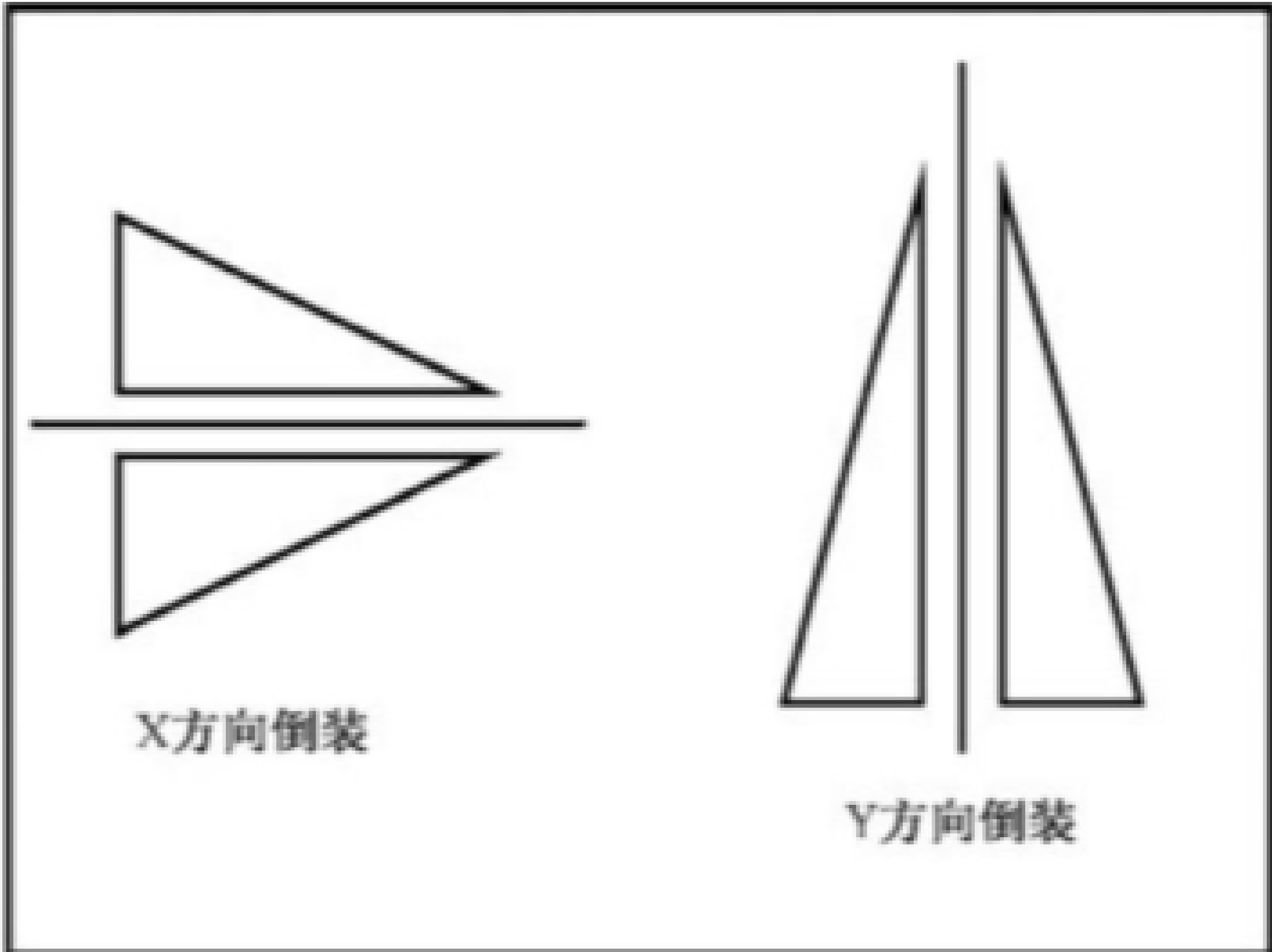


图 45 `X` 方向倒装和 `Y` 方向倒装示意图

- j) angle:该属性定义逆时针旋转相对于组件的参考点的角度,如果没有指定角度,则将零设置为默认值。旋转角度的单位由<unit>语句中的<angle>语句定义。
- k) mount:此属性定义组件的放置边,参数应为下列任何一项(参见图 46):
  - TOP 组件放在顶部;
  - BOTTOM 组件放在底部;
  - MIDDLE 组件嵌入基板内层。

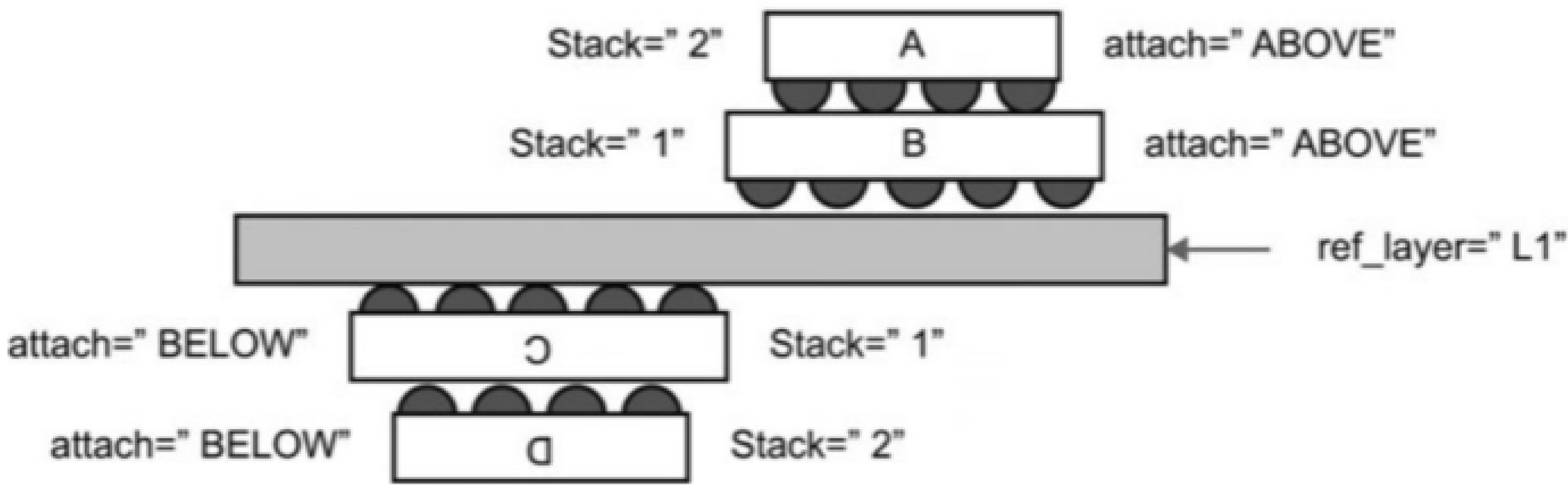


图 46 未指定层时组件放置的说明示例

- l) ref\_layer:此属性指定了安装层,即通过安装在该层上的另一个组件直接或间接安装组件的基板导电层。该属性通过 R 格式中<layer>语句的 name 属性定义的层名称指定安装层。
- m) attach:该属性与 ref\_layer 属性一起使用,并指定组件放置边。可用的关键字是“ABOVE”或“BELOW”(见图 42)。如果指定了“ABOVE”,则将组件放置在指定层的上方。如果指定了“BELOW”,则将组件放置在指定层的下方。如果未指定,则将“ABOVE”设置为默认值。
- n) stack:此属性指定一个称为堆叠级别的正整数,用于定义子组件之间 Z 方向的堆叠顺序。对于两个组件 A 和 B(C 和 D),如果以下所有条件都成立,则 A(D)被定义为在 Z 方向上高于(低于)B(C)(图 47):
  - 1) A 和 B 的安装属性相同或者 A 和 B 各自的 ref\_layer 属性和 attach 属性相同;
  - 2) A 和 B 的形状在 X-Y 平面上重叠;
  - 3) 如果安装属性为“TOP”或“MIDDLE”(BOTTOM),或者放置为“ABOVE”(BELOW),则 A 的堆叠级别比 B 的堆叠级别大(小)。

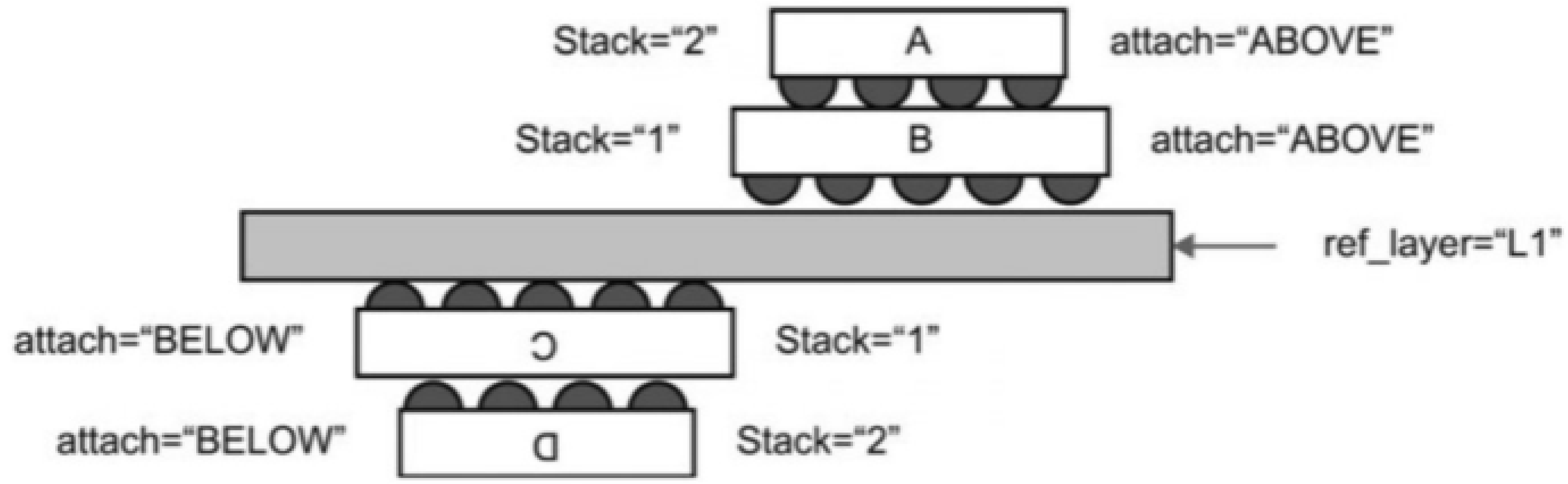


图 47 堆叠属性的示例图

图 47 的示例可用以下代码表示:

```
<placement ref_module="CAP1"inst="A"x="10"y="10"
  ref_layer="L1"attach="ABOVE"stack="2"/>
<placement ref_module="CAP2"inst="B"x="10"y="10"
  ref_layer="L1"attach="ABOVE"stack="1"/>
<placement ref_module="CAP3"inst="C"x="-10"y="-10"
```

```

        ref_layer="L1"attach="BELOW"stack="1"/>
<placement ref_module="CAP4"inst="D"x="-10"y="-10"
        ref_layer="L1"attach="BELOW"stack="2"/>
ref_rule_name

```

- o) `ref_rule_name`: 此属性指定放置的组件的区域的設計規則。設計規則由 R 格式中 `<Physicaldesign>` 语句的 `name` 属性引用。
- p) `sizing`: 此属性指定大小宽度, 该宽度是正(负)数, 用于将组件的形状放大(缩小)离值。

#### 8.2.10.4.3 例句

以下是 `<placement>` 语句的示例。

```

<component>
  <placement ref_module="CAP0603B"inst="C10"
    x="-8584.7"y="-4104.9"mount="BOTTOM"/>
  <placement ref_module="CAP0603B"inst="C11"
    x="-8584.7"y="-6355.9"mount="BOTTOM"/>
  <placement ref_module="CAP1005B"inst="C12"
    x="26092.5"y="37686.8"angle="90"mount="BOTTOM"/>
  <placement ref_module="CAP1005B"inst="C13"
    x="30005.4"y="37686.8"angle="90"mount="BOTTOM"/>
  <placement ref_module="CAP1005B"inst="C14"
    x="34659.6"y="37686.8"angle="90"mount="BOTTOM"/>
  <placement ref_module="CAP1005B"inst="C15"
    x="40178.8"y="37686.8"angle="90"mount="BOTTOM"/>
  <placement ref_module="RAS8"inst="RAS8_RN2"
    x="25477.7"y="15729.6"angle="270"mount="TOP"/>
  <placement ref_module="RAS8"inst="RAS8_RN4"
    x="43929.3"y="6225.3"angle="270"mount="TOP"/>
  <placement ref_module="REGULATOR"inst="REGULATOR"
    x="-12598"y="10183.9"mount="TOP"/>
  <placement ref_module="SOC_PKG"inst="SOC"
    x="400"y="-6500"mount="TOP"/>
  <placement ref_module="XTAL"inst="XTAL"
    x="-6285.9"y="26473"mount="TOP"/>
</component>

```

#### 8.2.10.5 bondingwire 语句

##### 8.2.10.5.1 概述

`<bondingwire>` 用于定义键合线的位置、起点和终点、层以及键合线规则。

```

<bondingwire

```



```
bw_id="identifier "  
[bw_rule_name="name_referenced_bonding_wire_rule "  
[bw_shape="name_of_referenced_bondingwire_shape "  
x1="x_coordinate_of_start_point "  
y1="y_coordinate_of_start_point "  
inst1="instance_name_of_start_point "  
ref_layer1="name_for_reference_end_layer "  
x2="x_coordinate_of_end_point "  
y2="y_coordinate_of_end_point "  
inst2="instance_name_of_end_point "  
ref_layer2="name_of_reference_end_layer "  
[loop="loop_direction "  
  
/>
```

8.2.10.5.2 属性定义

〈bondingwire〉语句属性定义如下。

- a) bw\_id:此属性是一个字符串,用于定义键合线的标识(id)。id 在〈component〉语句中是唯一的。
- b) bw\_rule\_name:此属性指定键合线的设计规则。设计规则由 R 格式中〈Physicaldesign〉语句的 name 属性引用。
- c) bw\_shape:此属性指定键合线的形状。形状通过 R 格式中〈bondingwire\_def〉语句的 name 属性引用。
- d) x1/y1:这些属性指定了键合线起始点的坐标。坐标单位由〈unit〉语句中的〈distance〉语句定义。
- e) inst1/ref\_layer1:这些属性指定在键合线的起点连接的布局对象。如果键合线连接到子组件,则其实例名称由属性 inst1 引用。如果键合线连接到基板的层,则层由其名称指定,该名称使用 R 格式中定义的 ref\_layer1 属性。inst1 或 ref\_layer1 属性应都存在。
- f) x2/y2:这些属性指定了键合线终点的坐标。坐标单位由〈unit〉语句中的〈distance〉语句定义。
- g) inst/ref\_layer2:这些属性指定在键合线的终点连接的布局对象。如果键合线连接到子组件,则其实例名称由属性 inst2 引用。如果键合线连接到基板的层,则层由其名称指定,该名称使用 R 格式中定义的 ref\_layer2 属性。Inst2 或 ref\_layer2 属性应都存在。
- h) loop:该属性指定键合线在 Z 方向上是向上凸起还是向下凸起:
  - 1) TOP:键合线在 Z 方向上向上凸出;
  - 2) BOTTOM:键合线在 Z 方向上向下凸出。

8.2.10.5.3 例句

以下是〈bondingwire〉语句的示例,见图 48。

```
<component ref_rule_name="bondingwire_rule">  
  <bondingwire bw_id="WB1"bw_shape="WBOND1"  
    x1="4.0"y1="1.0"inst1="U1"  
    x2="6.0"y2="1.0"ref_layer2="L1"  
    loop="TOP"/>
```

```
<bondingwire bw_id="WB2"bw_shape="WBOND2"
  x1="1.0"y1="2.0"inst1="U2"
  x2="2.0"y2="2.0"inst2="U1"
  loop="TOP"/>
<bondingwire bw_id="WB3"
  bw_rule_name="bondingwire_rule_b"bw_shape="WBONDB"
  x1="3.0"y1="3.0"inst1="U3"
  x2="6.0"y2="3.0"ref layer2="L3"
  loop="BOTTOM"/>
</component>
```

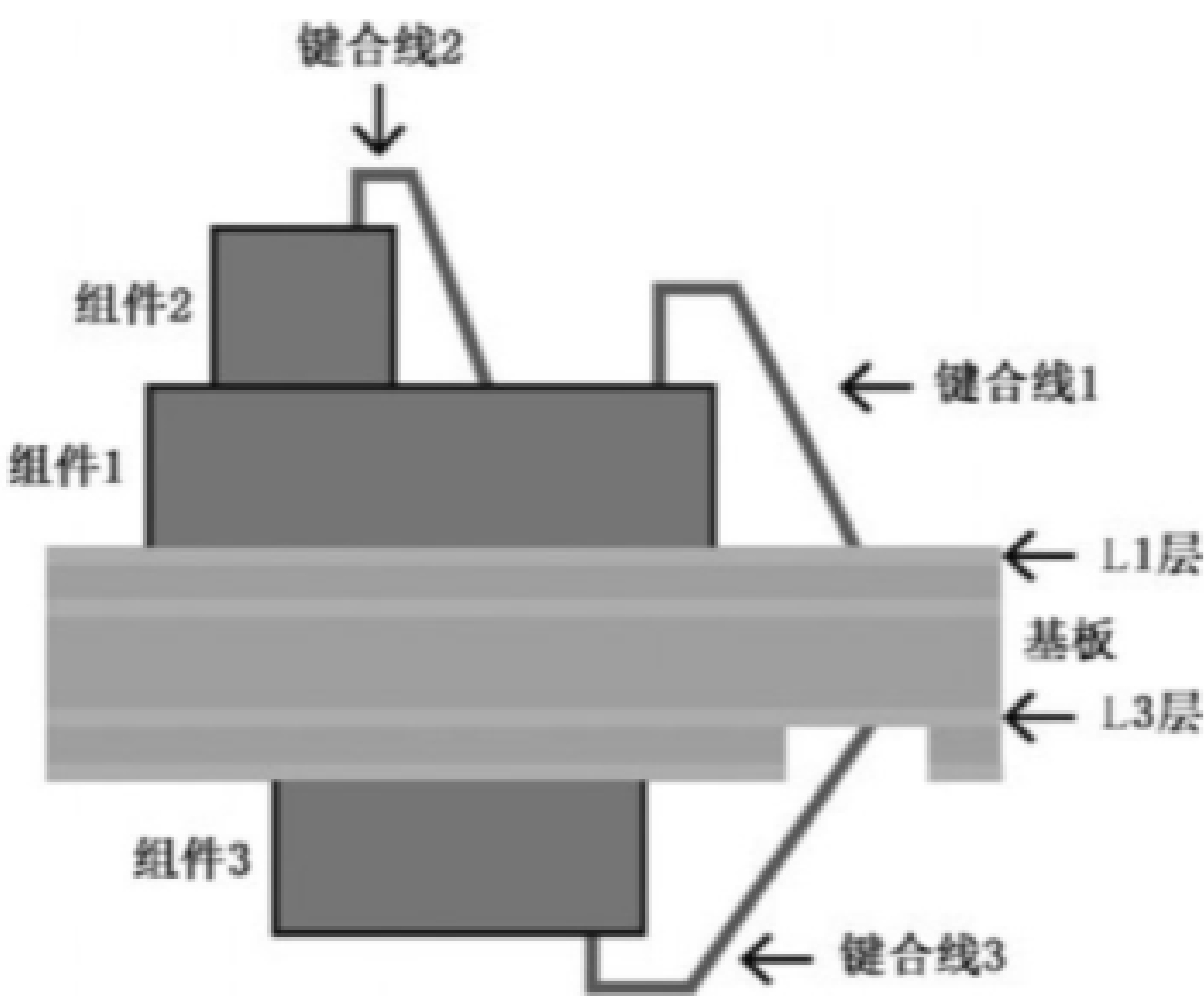


图 48 键合线示例图

8.2.11 keepaway 语句

8.2.11.1 概述

<keepaway>语句定义了该组件周围的一个区域,其他组件不应放置在该区域中。

```
<keepaway
  shape_id="identifier_of_referenced_shape "
  x="x_coordinate"
  y="y_coordinate "
  [angle="angel "]
  [objective_layer="keep_away_area "]
>
  [extensions> element]...
</keepaway>
```

8.2.11.2 属性定义

<keepaway>语句属性定义如下。

- a) shape\_id:此属性指定预定义形状的标识号。
- b) x/y: 这些属性指定形状原点相对于组件原点的坐标。坐标的单位由<unit>语句中的<distance>语句定义。

- c) angle:该属性指定形状的旋转角度。如果未指定角度,则设置零为默认值。旋转角度的单位由<unit>语句中的<angle>语句定义。
- d) objective\_layer:该属性指定 Z 方向上的有效范围:
  - 1) SAME:该属性对基板/层的同一安装侧有效;
  - 2) OPPOSITE:该属性对任意部分有效;
  - 3) ALL:该属性对所有部分有效。

8.2.11.3 语句组成

<keepaway>语句包含以下内容:  
<extensions>

8.2.11.4 例句

以下是<keepaway>语句的示例。

```
<keepaway shape_id="PFBAGBODY" x="0" y="0" objective_layer="ALL"/>
```

8.2.12 reference 语句

8.2.12.1 概述

<reference>语句用于建立 C 格式模块与电气、热力模块之间的关系、或几何模型文件。在引用电气或热力文件时,<reference>语句定义<socket>语句中的端口与引用文件中的端口之间的连接程序,同时参考几何文件,该语句以 C 格式将参考几何文件中的对象转换为模块。

```
<reference
{ xmlns:verilog="http://www.jeita.or.jp/LPB/verilog" |
  xmlns:VHDL="http://www.jeita.or.jp/LPB/VHDL" |
  xmlns:def="http://www.jeita.or.jp/LPB/def" |
  xmlns:spice="http://www.jeita.or.jp/LPB/spice" |
  xmlns:touchstone="http://www.jeita.or.jp/LPB/touchstone" |
  xmlns:dxfl="http://www.jeita.or.jp/LPB/dxf" |
  xmlns:gds="http://www.jeita.or.jp/LPB/gds" |
  xmlns:step="http://www.jeita.or.jp/LPB/step" |
  xmlns:sat="http://www.jeita.or.jp/LPB/sat" |
  xmlns:iges="http://www.jeita.or.jp/LPB/iges" |
  xmlns:ibis="http://www.jeita.or.jp/LPB/ibis" |
  xmlns:ibis="http://www.jeita.or.jp/LPB/ibis_pkg" |
  xmlns:ibis="http://www.jeita.or.jp/LPB/ibis_ebd" |
  xmlns:systemc="http://www.jeita.or.jp/LPB/ systemc " |
  xmlns:xfl="http://www.jeita.or.jp/LPB/xfl" |
  xmlns:xml="http://www.jeita.or.jp/LPB/xml" |
  xmlns:thermal_2resistor="http://www.jeita.or.jp/LPB/thermal_2resistor" |
  xmlns: thermal_delphi="http://www.jeita.or.jp/LPB/ thermal_delphi" |
  xmlns:jtam="http://www.jeita.or.jp/LPB/jtam " |
```

```

xmlns:JLPB="http://www.jeita.or.jp/LPB/JLPB" }
reffile="name_of_referenced_file "
format="file_format "
[type="file_type "]
[distance="unit_of_length "]
[scale="geometric_scale "]
[comment="comment_text "]
>
        [<connection> element]...
        [<affine_transformation> element]...
        [<material> element]...
        [<heat_source> element]...
        [<extensions> element]...
</reference>

```

有些格式没有 I/O 端口的概念,在这种情况下,与信号 I/O 位置相对应的坐标用于创建与 C 格式文件中的端口的关系。例如,<reference>语句可为没有物理信息的 IBIS 的 I/O 节点提供位置。

#### 8.2.12.2 属性定义

<reference>语句的属性定义如下:

```

xmlns:verilog="http://www.jeita.or.jp/LPB/verilog"
xmlns:VHDL="http://www.jeita.or.jp/LPB/VHDL"
xmlns:def="http://www.jeita.or.jp/LPB/def"
xmlns:spice=http://www.jeita.or.jp/LPB/spice
xmlns:touchstone=http://www.jeita.or.jp/LPB/ touchstone
xmlns:dxfl="http://www.jeita.or.jp/LPB/dxf"
xmlns:gds=http://www.jeita.or.jp/LPB/gds
xmlns:step=http://www.jeita.or.jp/LPB/ step
xmlns:sat=http://www.jeita.or.jp/LPB/ sat
xmlns:iges=http://www.jeita.or.jp/LPB/ iges
xmlns:ibis=http://www.jeita.or.jp/LPB/ibis
xmlns:ibis_pkg="http://www.jeita.or.jp/LPB/ibis_pkg "
xmlns:ibis_ebd=http://www.jeita.or.jp/LPB/ibis_ebd
xmlns:systemc ="http://www.jeita.or.jp/LPB/ systemc "
xmlns:xfl="http://www.jeita.or.jp/LPB/xfl"
xmlns:xml="http://www.jeita.or.jp/LPB/xml"
xmlns:thermal_2resistor="http://www.jeita.or.jp/LPB/ thermal_2resistor"
xmlns: thermal_delphi ="http://www.jeita.or.jp/LPB/ thermal_delphi "
xmlns:jtam="http://www.jeita.or.jp/LPB/ jtam "
xmlns:JLPB="http://www.jeita.or.jp/LPB/JLPB"
xmlns:extensions="http://www.jeita.or.jp/LPB/ extensions "

```

这一行是可扩展标记语言(XML)[B2]<sup>6</sup>的命名空间,命名空间由引用文件的格式正确使用,相应的

命名空间和文件格式如下所示。

- a) `xmlns:verilog`: 硬件描述语言, 用于门级或寄存器级数字电路的设计与验证。
- b) `xmlns:VHDL`: 超高速集成电路硬件描述语言(VHDL), 用于数字和混合信号系统的设计。
- c) `xmlns:def`: 设计交换格式(DEF), 以美国信息交换标准编码(ASC II)格式表示集成电路的物理布局, 它表示网表、元器件放置和路由信息。
- d) `xmlns:spice`: 标准 SPICE 是 Berkeley SPICE, 但是 SPICE-like 的电路模拟器并不完全兼容。使用 `type` 属性可以显示 SPICE 的指定版本, 为 SPICE 属性指定电路模拟器的名称, 为其保留 Berkeley\_SPICE 和 TBIS-ISS; 如果省略了 `type` 属性, 则表示 SPICE 是 Berkeley SPICE。  
`xmlns` 的 `type` 属性: `spice`  
Berkeley\_SPICE    伯克利 SPICE  
IBIS-ISS            IBIS 互连 SPICE 子电路  
*any\_string*        指定 SPICE-like 的电路联播器的名称
- e) `xmlns:touchstone`: 支持 Touchstone 版本 1.0 和 2.0。使用类型属性可以指定在 Touchstone 文件中定义的模型类型, 应是 S-参数。如果省略了 `type` 属性, 则表示在 Touchstone 文件中定义的模型是 S-参数。  
`xmlns` 的 `type` 属性: `touchstone`  
SParameter        S 参数模型
- f) `xmlns:dxf`: 图形转换格式(DXF), 它用于矢量图像文件。
- g) `xmlns:gds`: GDS II 流格式, 它是一种数据库文件格式, 用于集成电路或集成电路图的数据交换。
- h) `xmlns:step`: ISO 10303, 用于在计算机辅助设计(CAD)工具中表示三维对象。
- i) `xmlns:sat`: 标准 ACIS 文本文件, 用于保存 ACIS 建模器的数据。
- j) `xmlns:iges`: 初始图形转换规范, 用于将三维对象表示为曲面模型的组合。
- k) `xmlns:ibis`: 输入/输出缓冲区信息规范(IBIS), 集成电路供应商使用它向客户提供有关产品 I/O 缓冲区的信息。
- l) `xmlns:ibis_pkg`: IBIS 包模型文件, 集成电路供应商使用它向客户提供有关产品 I/O 缓冲区的信息。
- m) `xmlns:ibis_ebd`: IBIS 电路板描述文件, 集成电路供应商使用它向客户提供有关产品 I/O 缓冲区的信息。
- n) `xmlns:systemc`: 此名称空间用于引用 SystemC。SystemC 用于系统和硬件设计。
- o) `xmlns:xml`: 此名称空间用于引用 XML 语言定义的外部文件, 如国际电工委员会(IEC)集成电路发射模型(ICEM)或集成电路免疫模型(ICIM)文件。

使用 `type` 属性可以指定在 XML 中定义的模型文件的类型。类型应为以下之一:

- `xmlns` 的 `type` 属性: `xml`;
- ICEM-CE: 集成电路模拟传导发射(IEC 62433-2[B5]);
- ICEM-RE: 集成电路电气模型辐射发射(IEC 62433-3[B6]);
- ICIM-CI: 集成电路抗扰度模式传导抗扰度(IEC 62433-4[B7]);
- CICM-RI: 集成电路抗扰度模型辐射抗扰度(IEC 62433-5[B8]);
- CEML: 传导发射标记语言;
- REML: 辐射发射标记语言;
- CILM: 传导抗扰度标记语言;
- RIML: 辐射免疫标记语言;
- PIML: 脉冲免疫标记语言。

- p) `xmlns:thermal_2resistor`: 此名称空间用于参考 JEDEC 2-电阻器紧凑型热模型。
- q) `xmlns:thermal_delphi`: 该名称空间用于参考 JEDEC DELPHI 紧凑型热模型。
- r) `xmlns:jtam`: 此名称空间用于引用 JTAM 文件。
- s) `xmlns:xfl:LPB G` 格式。
- t) `xmlns:JLPB:LPB C` 格式。
- u) `xmlns:extensions`: 此命名空间用于使用 `<extensions>` 语句的模型文件。
- v) `reffile`: 此属性定义要与其建立关系的文件的名称。
- w) `format`: 此属性定义引用文件的语言。参数应为下列之一:
  - VERILOG: 硬件描述语言;
  - VHDL: VHDL 语言文件;
  - DEF: 设计交换格式 (DEF);
  - SPICE: 以集成电路为重点的仿真程序的网表;
  - DXF: 图像转换格式 (DXF);
  - GDS: GDS II 流格式;
  - IBIS: 输入/输出缓冲区信息规范;
  - XML: XML 语言文件;
  - XFL: LPB G 格式;
  - JLPB: LPB C 格式。
- x) `type`: 此属性与 `xmlns:spice`, `xmlns:touchstone`, 和 `xmlns:xml` 属性一起使用。在 `xmlns:spice` 的情况下, 它指定 SPICE 的电路模拟器类型。Berkeley\_SPICE 和 IBIS-ISS 是为其保留的, 如果省略 `type` 属性, 则表示 SPICE 是 Berkeley SPICE。可用值如下:
  - 1) Berkeley\_SPICE: Berkeley\_SPICE;
  - 2) IBIS-ISS: IBIS 互连 ISS 子电路;
  - 3) `any_string`: 规定类 SPICE 电路模拟器名称。
 在 `xmlns:touchstone` 的情况下, 它指定了标准文件的模型类型。可用值为 SParameter。如果省略了 `type` 属性, 则表示标准文件中定义的模型是 S 参数。  
 SParameter: S 参数模型。
 对于 `xmlns:xml`, 它指定由 XML 定义的模型文件的类型。属性值应为下列值之一:
  - 1) ICEM-CE: 集成电路电子模型—传导发射 (IEC 62433-2[B5]);
  - 2) ICEM-RE: 集成电路电子模型—辐射发射 (IEC 62433-3[B6]);
  - 3) ICIM-CI: 集成电路抗扰度模型—传导抗扰度 (IEC 62433-4[B7]);
  - 4) ICIM-RI: 集成电路抗扰度模型—辐射抗扰度 (IEC 62433-3[B8]);
  - 5) CEML: 传导发射标记语言;
  - 6) REML: 辐射发射标记语言;
  - 7) CIML: 传导抗扰度标记语言;
  - 8) RIML: 辐射抗扰度标记语言;
  - 9) PIML: 脉冲抗扰度标记语言。
- y) `distance`: 此属性指定引用文件的距离单元系统, 如果未指定此属性, 则使用引用文件的默认值。参数为下列之一:
  - pm: 皮米;
  - nm: 纳米;
  - $\mu$ m: 微米;
  - mm: 毫米;

m:米。

- z) scale:此属性指定引用文件的刻度范围,刻度范围应大于零。如果未指定此属性,则使用 1.0 作为默认属性。

8.2.12.3 语句组成

<reference>语句包含以下内容:

- <connection>
- <affine\_transformation>
- <material>
- <heat\_source>
- <extensions>

8.2.12.4 connection 语句

8.2.12.4.1 概述

<connection>语句定义了<socket>语句中定义的<port>与引用文件的 I/O 终端之间的关系。

```
<connection
  socket_name="name_of_reference_socket "
  { port_name="name_of_referenced_port " |
    port_id="identifier_of_referenced_port_id " }
>

  [<verilog:ref_port> element]...
  [<VHDL:ref_port> element]...
  [<def:ref_port> element]...
  [<spice:ref_port> element]...
  [<touchstone:ref_port> element]...
  [<dxfl:ref_port> element]...
  [<gds:ref_port> element]...
  [<xfl:ref_port> element]...
  [<ibis:ref_port> element]...
  [<ibis_pkg:ref_port> element]...
  [<ibis_ebd:ref_port> element]...
  [<systemc:ref_port> element]...
  [<thermal_2resistor:ref_port> element]...
  [<thermal_delphi:ref_port > element]...
  [<jtam:ref_port > element]...
  [<JLPB:ref_port> element]...
  [<xml:ref_port> element]...
  [<extension> element]...
</connection>
```

连接方案取决于参考文件的设计语言,以下是一个典型的例子。  
将端口明确定义为硬件描述语言,使用“组件名”和“端口名”的组合来定义关系:

```
<connect socket_name="socket1" port_id="A1">  
  <verilog:ref_port module="topmodule" portname="DQ1"/>  
</connect>
```

在本例中,底座 1 中的 A1 端口与硬件描述语言的顶层组件中的终端 DQ1 相关联。  
对于以集成电路为重点的仿真程序(SPICE)的网表,使用 I/O 节点描述的子电路名称和顺序来定义关系:

```
<connect socket_name="socket1" port_id="A1">  
  <spice:ref_port subckt="spicetop" portid="5"/>  
</connect>
```

在本例中,底座 1 中的 A1 端口与子电路 spicetop 中以第五定义的 I/O 节点相关联。

8.2.12.4.2 定义属性

- <connection>语句的属性定义如下:
- a) socket\_name:此属性定义底座的名称,该底座包含与引用文件建立关系的端口,指定的底座应定义在相同的<socket>语句的<module>语句中;
  - b) port\_name:此属性定义要与引用文件建立关系的端口的名称,指定的端口应在<socket>语句中的<port>语句中定义,该语句由 socket\_name 属性指定,port\_name 属性不应与 port\_id 属性一起使用;
  - c) port\_id:此属性定义与引用文件建立关系的端口的标识符,指定的端口应在<socket>语句的<port>语句中定义,该语句由 socket\_name 属性指定,port\_id 属性不应与 port\_name 属性一起使用。

8.2.12.4.3 语句组成

<connection>语句包含以下内容:

```
<verilog:ref_port>  
<VHDL:ref_port>  
<def:ref_port>  
<spice:ref_port>  
<touchstone:ref_port>  
<dxfl:ref_port>  
<gds:ref_port>  
<xfl:ref_port>  
<ibis:ref_port >  
<ibis_pkg:ref_port>  
<ibis_ebd:ref_port>  
<systemc:ref_port>
```



```
<thermal_2resistor:ref_port>  
<thermal_delphi:ref_port>  
<jtam:ref_port>  
<JLPB:ref_port>  
<xml:ref_port>  
<extensions>
```

8.2.12.4.4 verilog:ref\_port 语句

8.2.12.4.4.1 概述

<verilog:ref\_port>语句用于与硬件描述语言文件建立关系。

```
<verilog:ref_port  
    module="module_name_in_verilog_file "  
    portname="port_name_in_verilog_file "  
>  
    [<extensions>]...  
</verilog:ref_port>
```

在硬件描述语言文件的情况下,通过组件名和端口名的组合创建关系。

8.2.12.4.4.2 定义属性

<verilog:ref\_port>的属性定义如下:

- a) module:此属性定义引用硬件描述语言文件中组件的名称;
- b) portname:此属性定义引用硬件描述语言文件中端口的名称,端口将在组件属性指定的组件中定义。

8.2.12.4.4.3 例句

以下是<verilog:ref\_port>语句的一个示例。

```
<reference  
    xmlns:verilog="http://www.jeita.or.jp/LPB/verilog"  
    reffile="XXXX.ver"  
    format="VERILOG"  
>  
    <connection socket_name="socket1" port_id="A1">  
        <verilog:ref_port module="topmodule" portname="DQ1"/>  
    </connection>  
    <connection socket_name="socket1" port_id="A2">  
        <verilog:ref_port module="topmodule" portname="DQ2" />  
    </connection>  
    <connection socket_name="socket1" port_id="A3">  
        <verilog:ref_port module="topmodule" portname="DQ3" />
```

```
</connection>
</reference>
```

```
[XXXX.ver]
  module topmodule(DQ1,DQ2,DQ3)
```

8.2.12.4.5 VHDL:ref\_port 语句

8.2.12.4.5.1 概述

〈VHDL:ref\_port〉语句用于与用 VHDL 语言描述的模型文件建立关系。

```
<VHDL:ref_port
  entity="entity_name_in_VHDL_file "
  portname="port_name_in_VHDL_file "
  [subtype="subtype_of_in_VHDL_file "]
>
  [〈extensions〉]...
</VHDL:ref_port>
```

8.2.12.4.5.2 定义属性

〈VHDL:ref\_port〉语句的属性定义如下。

- a) entity:此属性定义引用 VHDL 文件中实体的名称。
- b) portname:此属性定义引用 VHDL 文件中端口的名称,端口应在实体属性指定的实体中定义。
- c) Subtype:此属性指定 VHDL 文件中端口声明的子类型。如果未指定此属性,则将电气设置为默认值:
  - 1) Electrical:参考端口用于电气连接;
  - 2) Thermal:参考端口用于以开尔文为单位的散热网络;
  - 3) thermal\_c:参考端口用于以摄氏度为单位的散热网络;
  - 4) any\_string:指定端口的子类型。

8.2.12.4.5.3 例句

以下是〈VHDL:ref\_port〉语句的一个示例。

```
<reference
  xmlns:VHDL="http://www.jeita.or.jp/LPB/VHDL"
  reffile="XXXX.vhd"
  format="VHDL"
>
  <connection socket_name="socket1" port_id="A1">
    <VHDL:ref_port entity="topmodule" portname="DQ1"/>
  </connection>
  <connection socket_name="socket1" port_id="A2">
    <VHDL:ref_port entity="topmodule" portname="DQ2" />
```

```
</connection>
<connection socket_name="socket1" port_id="A3">
  <VHDL:ref_port entity="topmodule" portname="DQ3" />
</connection>
</reference>
[XXXX.ver]
entity topmodule is
port
(
  DQ1      : out    std_logic;
  DQ2      : out    std_logic;
  DQ3      : out    std_logic;
);
end topmodule;
```

8.2.12.4.6 def:ref\_port 语句

8.2.12.4.6.1 概述

<def:ref\_port>语句用于与 DEF 文件建立关系。

```
<def:ref_port
  [comp="component_name_in_DEF_file "]
  pinname="pin_name_in_DEF_file "
>
  [<extensions>element]...
</def:ref_port>
```

对于 DEF 文件,使用引脚名和/或元器件名创建关系。

8.2.12.4.6.2 定义属性

<def:ref\_port>语句的属性如下所示:

- a) comp:此属性定义在 DEF 文件中的 Component 部分中定义的元器件的名称;
- b) pinname:此属性定义在引用 DEF 文件的 PINS 部分中定义的引脚的名称,或者它指定由 comp 属性定义的元器件的引脚名。

8.2.12.4.6.3 例句

以下是<def:ref\_port>语句的一个示例。

```
<reference
  xmlns:def="http://www.jeita.or.jp/LPB/def"
  reffile="XXXX.def"
  format="DEF"
>
```

```

    <connection socket_name="socket1" port_id="A1">
      <def:ref_port comp="SBIO1" pinname="Z"/>
    </connection>
    <connection socket_name="socket1" port_id="A2">
      <def:ref_port comp="SBIO2" pinname="Z"/>
    </connection>
    <connection socket_name="socket1" port_id="A3">
      <def:ref_port comp="SBIO2" pinname="Z"/>
    </connection>
  </reference>

```

[XXXX.def]

```

    COMPONENTS    100 ;
-   SBIO1    io    ;
-   SBIO2    io    ;
-   SBIO3    io    ;

    .....
END COMPONENTS

```

```

<reference xmlns:def=http://www.jeita.or.jp/LPB/def
  reffile="YYY.def"
  format="DEF"
>
  <connection socket_name="socket1" port_id="A1">
    <def:ref_port pinname="PIN1"/>
  </connection>
  <connection socket_name="socket1" port_id="A2">
    <def:ref_port pinname="PIN2"/>
  </connection>
  <connection socket_name="socket1" port_id="A3">
    <def:ref_port pinname="PIN3"/>
  </connection>
</reference>

```

[YYYYY.def]

```

    PINS 10    ;
-   PIN1    oDQ1    ;
-   PIN2    oDQ2    ;
-   PIN3    oDQ3    ;

    ...
END PINS

```

8.2.12.4.7 spice:ref\_port 语句

8.2.12.4.7.1 概述

⟨spice:ref\_port⟩语句用于与以集成电路为重点的仿真程序的(SPICE)网表文件建立关系。

```
⟨spice:ref_port
    subckt="name_of_subckt "
    portid="order_of_pins_in_subckt "
>
    [⟨extensions⟩element]...
⟨/spice:ref_port⟩
```

在 SPICE 的情况下,通过 I/O 节点描述的子电路名称和顺序的组合来创建关系。对于以集成电路为重点的仿真程序的网表文件,使用.subckt 子电路(.subckt)的名称和.subckt 行上 I/O 节点的顺序来定义关系。

8.2.12.4.7.2 定义属性

- ⟨spice:ref\_port⟩语句的属性定义如下:
- a) subckt:此属性定义引用 SPICE 文件中子电路(.subckt)的名称;
  - b) portid:此属性定义.subckt 行中 I/O 节点的顺序,该值应为 1 或更多的整数。

8.2.12.4.7.3 例句

以下是⟨spice:ref\_port⟩语句的一个示例。

```
⟨reference
    xmlns:spice="http://www.jeita.or.jp/LPB/spice"
    reffile="XXXX.sp"
    format="SPICE"
    type="HSPICE "
>
    <connection socket_name="socket1" port_id="A1">
        <spice:ref_port subckt="top" portid="3"/>
    </connection>
    <connection socket_name="socket1" port_id="A2">
        <spice:ref_port subckt="top" portid="2"/>
    </connection>
    <connection socket_name="socket1" port_id="A3">
        <spice:ref_port subckt="top" portid="1"/>
    </connection>
</reference>

[XXXXX.sp]
```

```
Subckt top p1 p2 p3 ;
```

SPICE 模型文件包括 IBIS-ISS。若要引用 IBIS-ISS 模型,在 type 属性中指定 IBIS-ISS 关键字,如下所示。

```
<reference
  xmlns:spice=http://www.jeita.or.jp.LPB/spice
  reffile="YYYY.ibs"
  format="SPICE"
  type="IBIS-ISS "
>
  <connection socket_name="socket" port_id="A1">
    <spice:ref_port subckt="top" portid="3" >
  </connection>
.....
```

8.2.12.4.8 <touchstone:ref port> 语句

8.2.12.4.8.1 概述

<touchstone:ref\_port>语句用于与 S 参数文件建立关系。

```
<touchstone:ref_port
  portid="order_of_ports_in_touchstone "
>
  [<extensions> element]...
/touchstone:ref_port>
```

8.2.12.4.8.2 属性定义

<touchstone:ref\_port>语句的属性定义如下：  
portid:此属性指定 S 参数文件中端口的顺序。该值应为 1 或更大的整数。

8.2.12.4.8.3 例句

以下是<touchstone:ref\_port>语句的一个示例。

```
<reference
  xmlns:touchstone="http://www.jeita.or.jp/LPB/touchstone"
  reffile="XXXX.s2p"
  format="TOUCHSTONE"
  type="SParameter"
>
  <connection socket_name="YYY"port_id="A">
    <touchstone:ref_port portid="1"/>
  </connection>
```

```
    <connection socket_name="YYY"port_id="B">
      <touchstone:ref port portid="2"/>
    </connection>
  </reference>
```

8.2.12.4.9 dxf:ref\_port 语句

8.2.12.4.9.1 概述

<dxf:ref\_port>语句用于与 DXF 文件建立关系。

```
<dxf:ref_port
    x="x_coordinate "
    y="y_coordinate "
    dxf_layer="layer_name_in_DXF_file "
    mount="mount_type "
>
    [ <extensions>element]...
</dxf:ref_port>
```

DXF 语言没有 I/O 终端的概念,因此,坐标用于在 C 格式文件中创建与端口的关系。

8.2.12.4.9.2 定义属性

<dxf:ref\_port>语句的属性定义如下。

- a) x/y:此属性定义与信号输入和输出点位置相对应的坐标。
- b) dxf\_layer:此属性定义放置信号输入和输出点的层的名称。
- c) mount:此属性定义信号输入和输出点的放置位置,参数应为下列任何一项:
  - 1) TOP:I/O 点放置在层的顶部;
  - 2) BOTTOM:I/O 点放置在层的底部。

8.2.12.4.9.3 例句

以下是<dxf:ref\_port>语句的一个示例。

```
<reference
  xmlns:dxf="http://www.jeita.or.jp/LPB/dxf"
  reffile="XXXX.dxf"
  format="DXF"
  distance="mm"
>
  <connection socket_name="socket1" port_id="A1">
    <dxf:ref_port x="100" y="8978" dxf_layer="L1" module="TOP" />
  </connection>
  <connection socket_name="socket1" port_id="A2">
    <dxf:ref_port x="200" y="8978" dxf_layer="L1" module="TOP" />
  </connection>
```

```

    <connection socket_name="socket1" port_id="A3">
      <dxs:ref_port x="300" y="8978" dxf_layer="L1" module="TOP" />
    </connection>
  </reference>

```

#### 8.2.12.4.10 gds:ref\_port 语句

##### 8.2.12.4.10.1 概述

〈gds:ref\_port〉语句用于与 GDS II 流格式文件建立关系。

```

<gds:ref_port
  x="x_coordinate "
  y="y_coordinate "
  gdx_layer="layer_number_in_GDS_file "
>
  [〈extensions〉element]...
</gds:ref_port>

```

GDS II 语言没有 I/O 终端的概念,因此,坐标用于在 C 格式文件中创建与端口的关系。

##### 8.2.12.4.10.2 定义属性

〈gds:ref\_port〉语句的属性定义如下:

- a) x/y:此属性定义与信号输入和输出点位置相对应的坐标,坐标的单位由〈unit〉语句中的〈distance〉语句定义;
- b) gds\_layer:此属性定义放置信号输入和输出点的层数。

##### 8.2.12.4.10.3 例句

以下是〈gds:ref\_port〉语句的一个示例。

```

<reference
  xmlns:gds="http://www.jeita.or.jp/LPB/gds"
  reffile="XXXX.gds"
  format="GDS"
>
  <connection socket_name="socket1" port_id="A1">
    <gds:ref_port x="100" y="8978" gds_layer="L1" />
  </connection>
  <connection socket_name="socket1" port_id="A2">
    <gdsref_port x="200" y="8978" gds_layer="L1" />
  </connection>
  <connection socket_name="socket1" port_id="A3">
    <gds:ref_port x="300" y="8978" gds_layer="L1" />
  </connection>

```



</reference>

8.2.12.4.11 xfl:ref\_port 语句

8.2.12.4.11.1 概述

<xfl:ref\_port>语句用于与 LPB G 格式文件建立关系。

```
<xfl:ref_port
    component="component_name_in_XFL_file "
    pinname="pin_name_in_XFL_file "
>
    [<extensions>element]...
</xfl:ref_port>
```

8.2.12.4.11.2 定义属性

<xfl:ref\_port>语句的属性定义如下：

- a) component:此属性定义引用 G 格式文件中.component 部分中定义的元器件的名称；
- b) pinname:此属性定义在引用 G 格式文件的部件部分中定义的元器件的引脚名称。

8.2.12.4.11.3 例句

以下是<xfl:ref\_port>语句的一个示例。

```
<reference
    xmlns:xfl="http://www.jeta.or.jp/LPB/xfl"
    reffile="XXXX.xfl"
    format="XFL"
>
<connection socket_name="socket1" port_id="A1">
    <xfl:ref_port component="NEWLSIDIE" pinname="1" />
</connection>
<connection socket_name="socket1" port_id="A2">
    <xfl:ref_port component="NEWLSIDIE" pinname="2" />
</connection>
<connection socket_name="socket1" port_id="A3">
    <xfl:ref_port component="NEWLSIDIE" pinname="3" />
</connection>
</reference>
```

```
[XXXX.xfl]
.part
DIE R -13.5 -13.5 13.5 13.5 0 S 0 {
1 -4 -5 B 4
2 -3 -5 B 4
```

```
3 -2 -5 B 4
      :
}
.end part
.component
    NEWLSIDIE    DIE    0 0 1 0
.end component
```

8.2.12.4.12 **ibis:ref\_port** 语句

8.2.12.4.12.1 概述

⟨ibis:ref\_port⟩语句用于与 IBIS 文件建立关系。

```
⟨ibis:ref_port
    component="component_name_in_IBIS_file "
    {signal_name="signal_name_in_IBIS_file " |
    pin_name="pin_name_in_IBIS_file "}
    [terminal_type="terminal_type in IBIS file "]
⟩
    [⟨extensions⟩element]...
⟨/ibis:ref_port⟩
```

8.2.12.4.12.2 定义属性

⟨ibis:ref\_port⟩语句的属性定义如下。

- a) component: 此属性指定在引用 IBIS 文件中的 [Component] 行中定义的元器件的名称。
- b) signal\_name: 此属性指定在引用 IBIS 文件中定义的 signal\_name。
- c) pin\_name: 此属性指定在参考 IBIS 文件中定义的端号名称。
- d) terminal\_type: 此属性指定在 IBIS 版本 7 或更高版本的互连模型中使用的端子类型。该值应为以下值之一：
  - 1) Pin\_I/O: 包装处的信号端子；
  - 2) Pad\_I/O: 芯片焊盘上的信号端子；
  - 3) Buffer\_I/O: I/O 缓冲器上的信号端子；
  - 4) Pin\_Rail: 包装处的电源端子；
  - 5) Pad\_Rail: 管芯焊盘上的电源端子；
  - 6) Buffer\_Rail: I/O 缓冲器处的电源端子；
  - 7) Pullup\_ref: I/O 缓冲器处的电源端子；
  - 8) Pulldown\_ref: I/O 缓冲器处的电源端子；
  - 9) Power\_clamp\_ref: I/O 缓冲器处的电源端子；
  - 10) Gnd\_clamp\_ref: I/O 缓冲器处的电源端子；
  - 11) Ext\_ref: I/O 缓冲器处的电源端子。如果未指定, 则将包处的终端设置为默认值。

8.2.12.4.12.3 例句

以下是⟨ibis:ref\_port⟩语句的示例。

示例 1：连接到封装终端，而不指定 terminal\_type 属性。

```
<reference
  xmlns:ibis="http://www.jeita.or.jp/LPB/ibis"
  reffile="XXXX.ibs"
  format="IBIS"
>
  <connection socket_name="socket1" port_id="A1">
    <ibis:ref_port component="DDR3-1Gbx16" signal_name="Vddq"/>
  </connection>
  <connection socket_name="socket1" port_id="A2">
    <ibis:ref_port component="DDR3-1Gbx16" signal_name="DQU5" />
  </connection>
  <connection socket_name="socket1" port_id="A3">
    <ibis:ref_port component="DDR3-1Gbx16" signal_name="DQU7" />
  </connection>
</reference>
```

示例 2：连接到指定 terminal\_type 属性的封装终端。

```
<reference
  xmlns:ibis="http://www.jeita.or.jp/LPB/ibis"
  reffile="XXXX.ibs"
  format="IBIS"
>
  <connection socket_name="socket1" port_id="A1">
    <ibis:ref_port component="DDR3-1Gbx16" signal_name="Vddq"
      terminal_type="Pin_Rail" />
  </connection>
  <connection socket_name="socket1" port_id="A2">
    <ibis:ref_port component="DDR3-1Gbx16" pin_name="A2"
      terminal_type="Pin_I/O" />
  </connection>
  <connection socket_name="socket1" port_id="A3">
    <ibis:ref_port component="DDR3-1Gbx16" pin_name="A3"
      terminal_type="Pin_I/O" />
  </connection>
</reference>
```

示例 3：连接到指定 terminal\_type 属性的裸片焊盘终端。

```
<reference
  xmlns:ibis="http://www.jeita.or.jp/LPB/ibis"
  reffile="XXXX.ibs"
```

```
format="IBIS"
>
  <connection socket_name="socket1" port_id="A1">
    <ibis:ref_port component="DDR3-1Gbx16" signal_name="Vddq"
      terminal_type="Pad_Rail "/ >
  </connection>
  <connection socket_name="socket1" port_id="A2">
    <ibis:ref_port component="DDR3-1Gbx16" pin_name="A2"
      terminal_type="Pin_I/O "/ >
  </connection>
  <connection socket_name="socket1" port_id="A3">
    <ibis:ref_port component="DDR3-1Gbx16" pin_name="A3"
      terminal_type=" Pin_I/O "/ >
  </connection>
</reference>
```

示例 4：连接到指定 terminal\_type 属性的 I/O 缓冲区终端。

```
<reference
  xmlns:ibis="http://www.jeita.or.jp/LPB/ibis"
  reffile="XXXX.ibs"
  format="IBIS"
>
  <connection socket_name="socket1" port_id="A1">
    <ibis:ref_port component="DDR3-1Gbx16" signal_name="Vddq"
      terminal_type=" Buffer_Rail "/ >
  </connection>
  <connection socket_name="socket1" port_id="A2">
    <ibis:ref_port component="DDR3-1Gbx16" pin_name="A2"
      terminal_type=" Buffer_I/O "/ >
  </connection>
  <connection socket_name="socket1" port_id="A3">
    <ibis:ref_port component="DDR3-1Gbx16" pin_name="A3"
      terminal_type=" Buffer_I/O "/ >
  </connection>
</reference>
```

```
XXXX.ibs
[IBIS ver]          7.0
[File name]         XXXXX.ibs
.....
[Component]         DDR3-1Gbx16
[Manufacturer]      JEITA_FACT
```

[Package]					
	variable	typ	min	max	
	R_pkg	0.5	0.4	0.6	
	L_pkg	1.5nH	1.0nH	2.0nH	
	C_pkg	0.4pF	0.2pF	0.6pF	
[Pin]	signal_name	model_name	R_pin	L_pin	C_pin
A1	Vddq	POWER			
A2	DQU5	DQ_MODEL	0.58	1.9nH	0.58pF
A3	DQU7	DQ_MODEL	0.57	1.85nH	0.57pF
.....					

8.2.12.4.13 ibis\_pkg:ref\_port 语句

8.2.12.4.13.1 概述

⟨ibis\_pkg:ref\_port⟩语句用于与 IBIS 封装模型文件建立关系。

```
⟨ibis_pkg:ref_port
    package_model="model_name_in_IBIS_pkg_file "
    pin_number="pin_number_in_IBIS_pkg_file "
>
    [⟨extensions⟩ element]..
⟨/ibis_pkg:ref_port⟩
```

8.2.12.4.13.2 属性定义

⟨ibis\_pkg:ref\_port⟩语句的属性定义如下：

- a) package\_model:此属性指定 IBIS 封装模型文件中[Define package model]描述的封装模型名称；
- b) pin\_number:此属性指定 IBIS 封装模型文件中[Pin numbers]数据的一个引脚号。

8.2.12.4.13.3 例句

以下是⟨ibis\_pkg:ref\_port⟩语句的一个示例。

```
⟨reference
    xmlns:ibis_pkg="http://www.jeita.or.jp/LPB/ibis_pkg"
    reffile="XXXX.ibs"
    format="IBIS_PKG"
>
    <connection socket_name="socket1"port_id="A1">
        <ibis_pkg:ref_port package_model="96ball_pkg"pin_number="A1"/></connection>
    <connection socket_name="socket1"port_id="A2">
        <ibis_pkg:ref_port package_model="96ball_pkg"pin_number="A2"/></connection>
    <connection socket_name="socket1"port_id="A3">
```

```
        <ibis_pkg:ref_port package_model="96ball_pkg" pin_number="A3"/>
    </connection>
</reference>
```

XXXX.ibs

```
[IBIS ver]          4.2
[File name]         XXXX.ibs
.....
[Component]        DDR3-1Gbx16
[Manufacturer]      JEITA_FACT
.....
[Define Package Model] 96ball_pkg
.....
[Number of Pins]    68
|
[Pin Numbers]
A1 | Vddq
A2 | Vss
A3 | DQU8
.....
```

8.2.12.4.14 **ibis\_ebd:ref\_port** 语句

8.2.12.4.14.1 概述

<ibis\_ebd:ref\_port>语句用于与 IBIS 电路板描述文件建立关系。

```
<ibis_ebd:ref_port
    board="board_name_in_IBIS_ebd_file "
    pin_name="pin_name_in_IBIS_ebd_file "
>
    [<extensions> element]..
</ibis_ebd:ref_port>
```

8.2.12.4.14.2 属性定义

<ibis\_ebd:ref\_port>语句的属性定义如下：

- a) board:此属性指定 IBIS 板描述文件中[Begin board Description]数据中的板名称；
- b) pin\_name:此属性指定 IBIS 板描述文件中[pin List]数据中的引脚名称之一。

8.2.12.4.14.3 例句

以下是<ibis\_ebd:ref\_port>语句的一个示例。

```
<reference
```

```
xmlns:ibis_ebd="http://www.jeita.or.jp/LPB/ibis_ebd"
reffile="XXXX.ebd"
format="IBIS_EBD"
>
  <connection socket_name="socket1"port_id="A1">
    <ibis_ebd:ref_port board="board"pin_name="1"/></connection>
  <connection socket_name="socket1"port_id="A2">
    <ibis_ebd:ref_port board="board"pin_name="2"/></connection>
  <connection socket_name="socket1"port_id="A3">
    <ibis_ebd:ref_port board="board"pin_name="3"/>
  </connection>
</reference>
```

```
XXXX.ebd
[IBIS ver]          4.2
[File name]         XXXXX.ebd
.....
[Begin Board Description]    board
[Manufacturer]    JEITA_FACT
|
|
[Number of Pins]    68
|
[Pin List]    signal_name
1            GND
2            GND
3            DQ04
.....
```

8.2.12.4.15 systemc:ref\_port 语句

8.2.12.4.15.1 概述

<systemc:ref\_port>语句用于与 C 系统文件建立关系。

```
<systemc:ref_port
    sc_module="module_name_in_systemc_file "
    portname ="port_name_of_function "
>
    [<extensions> element]..
</systemc:ref_port>
```

8.2.12.4.15.2 属性定义

<systemc:ref\_port>语句的属性定义如下：

- a) `sc_module`:此属性指定引用的 C 系统文件中 `sc_module` 的名称;
- b) `portname`:此属性指定引用的 C 系统文件中端口的名称。端口应在 `sc_module` 属性指定的 `sc_module` 中定义。

8.2.12.4.15.3 例句

以下是`<systemc:ref_port>`语句的一个示例。

```
<reference
  xmlns:systemc="http://www.jeita.or.jp/LPB/systemc"
  reffile="./sample_systemc.dll"
  format="SYSTEMC"
>
  <connection socket_name="pkg1"port_id="A1">
    <systemc:ref_port sc_module="and"portname="a[0]"/>
  </connection>
  <connection socket_name="pkg1"port_id="A2">
    <systemc:ref_port sc_module="and"portname="a[1]"/>
  </connection>
  <connection socket_name="pkg1"port_id="A3">
    <systemc:ref_port sc_module="and"portname="sum"/>
  </connection>
</reference>

//sample_systemc.hpp
#include "system.c"
SC_MODULE(and){
  sc_in<bool> a[2];//input port
  sc_out<bool>sum;//output port
  .....
};
//sample_systemc.cpp
#include "sample_systemc.hpp"

SC_MODULE_EXPORT(and);
```

8.2.12.4.16 `thermal_2resistor:ref_port` 语句

8.2.12.4.16.1 概述

`<thermal_2resistor:ref_port>` 语句用于与双电阻器型的热模型建立关系。可以定义 `socket` 名称,但无法定义端口。

```
<thermal_2resistor:ref_port
  model="model_name_in_thermal_model_file "
```



```
>
    [⟨extensions⟩ element]..
</thermal_2resistor:ref_port>
```

8.2.12.4.16.2 属性定义

⟨thermal\_2resistor:ref\_port⟩语句属性定义如下：  
model:该属性指定在双电阻器型的热模型文件中定义的模型名称。

8.2.12.4.16.3 例句

以下是⟨thermal\_2resistor:ref\_port⟩语句的一个示例。

```
<reference
  xmlns:thermal_2resistor="http://www.jeita.or.jp/LPB/thermal_2resistor"
  reffile="./thermal_model/sample_pkg_2resistor.tar.gz"
  format="THERMAL 2RESISTOR"
>
  <connection socket_name="sample_pkg">
    <thermal_2resistor:ref_port model="pkg_2resistor"/>
  </connection>
</reference>
```

8.2.12.4.17 thermal\_delphi:ref\_port 语句

8.2.12.4.17.1 概述

⟨thermal\_delphi:ref\_port⟩语句用于与德尔菲热模型建立关系。可以定义 socket 名称,但无法定义端口。

```
<thermal_delphi:ref_port
  model="model_name_in_thermal_model_file "
>
    [⟨extensions⟩ element]...
</thermal_delphi:ref_port>
```

8.2.12.4.17.2 属性定义

⟨thermal\_delphi:ref\_port⟩语句的属性定义如下：  
model:该属性指定在 DELPHI 热模型文件中定义的模型名称。

8.2.12.4.17.3 例句

以下是⟨thermal\_delphi:ref\_port⟩语句的一个示例。

```
<reference
  xmlns:thermal_delphi="http://www.jeita.or.jp/LPB/thermal_delphi"
  reffile="./thermal_model/sample_pkg_delphi.tar.gz"
```

```
format="THERMAL_DELPHI"  
>  
  <connection socket_name="sample_pkg">  
    <thermal_delphi:ref_port model="pkg_delphi"/>  
  </connection>  
</reference>
```

8.2.12.4.18 JLPB:ref\_port 语句

8.2.12.4.18.1 概述

<JLPB:ref\_port>语句用于与 LPB C 格式文件建立关系。

```
<JLPB:ref_port  
  module="module_name_in_LPB_CFormat "  
  socket="socket_name_in_LPB_CFormat "  
  [port_name="port_name_in_LPB_CFormat " |  
  port_id="port_identifier_in_LPB_CFormat " ]  
>  
  [ <extensions>element ] ...  
</JLPB:ref_port>
```

8.2.12.4.18.2 定义属性

<JLPB:ref\_port>语句的属性定义如下：

- a) module:此属性定义在引用 C 格式文件中的<module>语句中定义的组件的名称；
- b) socket:此属性定义在引用 C 格式文件中的<socket>语句中定义的底座的名称，底座应在 module 属性指定的组件中定义；
- c) port\_name:此属性定义在引用 C 格式文件中的<port>语句中定义的端口的名称，端口将在 socket 属性指定的底座中定义，port\_name 属性不应与 port\_id 属性一起使用。
- d) port\_id:此属性定义在引用 C 格式文件中的<port>语句中定义的端口的标识符，端口将在 socket 属性指定的底座中定义，port\_id 属性不应与 port\_name 属性一起使用。

8.2.12.4.19 xml:ref\_port 语句

8.2.12.4.19.1 概述

<xml:ref\_port>语句用于与 XML 语言描述的模型文件建立关系。

```
<xml:ref_port  
  port_path="xpath "  
>  
  [ <extensions>element ] ...  
</xml:ref_port>
```

模型文件中的节点路径使用 XML 路径语言(xpath)表示。

8.2.12.4.19.2 定义属性

<xml:ref\_port>语句的属性定义如下：  
 port\_path:此属性定义引用模型文件中的 I/O 节点的路径,路径由 XML 路径语言版本 1.0 表示。

8.2.12.4.19.3 例句

以下是<xml:ref\_port>语句的一个示例。

```

<reference
  xmlns:XML="http://www.jeita.or.jp/LPB/xml"
  reffile="ICEMCE.xml"
  format="XML"
  type="ICEM-CE"
>
  <connection socket_name="socket1" port_id="A1">
    <xml:ref_port port_path="/Cemodel/Lead_definitios/Lead[@Id='3']"/>
  </connection>
  <connection socket_name="socket1" port_id="A2">
    <xml:ref_port port_path="/Cemodel/Lead_definitios/Lead[@Id='4']"/>
  </connection>
  <connection socket_name="socket1" port_id="A3">
    <xml:ref_port port_path="/Cemodel/Lead_definitios/Lead[@Id='5']"/>
  </connection>
</reference>

[ICEMCE.xml]
<Cemodel>
  ...
  <Lead_definitions>

    <Lead Id="3" name="Vssq" Mode="GND"/>
    <Lead Id="4" name="DQU5" Mode="external"/>
    <Lead Id="5" name="DQU7" Mode="external"/>
    ...
  </Lead_definitions>
  ...
</Cemodel>
  
```

8.2.12.5 affine\_transformation 语句

8.2.12.5.1 概述

<affine\_transformation>语句定义了仿射空间变换矩阵,将参考数据文件中对象的坐标转换为 C 格式文件中的坐标。

```
<affine_transformation
    a11="real_number "
    a12="real_number "
    a13="real_number "
    a14="real_number "
    a21="real_number "
    a22="real_number "
    a23="real_number "
    a24="real_number "
    a31="real_number "
    a32="real_number "
    a33="real_number "
    a34="real_number "
>
    [<step:ref_product> element |
    <sat:ref_body> element]...
    [<extensions> element]...
</affine_transformation>
```

8.2.12.5.2 属性定义

$a_{11} \sim a_{34}$  是仿射空间变换矩阵的矩阵元素,如下所示。

$$M = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

因此,参考文件(x,y,z)中的坐标系到 C 格式(x',y',z')的坐标系的转换定义为:

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = M \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

8.2.12.5.3 语句内容

<affine\_transforming>语句包含以下内容。<step:ref\_product>和<sat:ref\_body>语句定义被引用文件中由仿射空间变换矩阵变换的对象。如果未定义任何对象,则会变换该文件中的所有对象。

```
<step:ref_product>
<sat:ref_body>
<extensions>
```

8.2.12.5.4 step:ref\_product 语句

8.2.12.5.4.1 概述

该语句用于指定 STEP 文件中通过仿射变换进行坐标系变换的对象。放置在指定对象的较低层次上的对象的坐标系也通过相同的仿射变换矩阵进行变换。通常,此属性指定引用 STEP 文件的顶级对

象。如果未指定对象,STEP 文件中的所有对象都将通过相同的仿射空间变换矩阵进行变换。

```
<step:ref_product
    name="objective_name_in_STEP_file "
>
    [<extensions> element]..
</step:ref_product>
```

8.2.12.5.4.2 属性定义

<step:ref\_product>语句的属性定义如下:  
name:该属性指定 STEP 文件中对象的名称,该对象的坐标系通过仿射变换进行变换。

8.2.12.5.4.3 例句

在以下示例中,IC1\_PACKAGE 和 IC2\_PACKAGE 对象的坐标在 X 方向上移动 100,在 Y 方向上移动 200。

```
<reference xmlns:step="http://www.jeita.or.jp/LPB/step"
    reffile="xxx.stp"format="STEP">
    <affine_transformation
        a11="1.0"a12="0.0"a13="0.0"a14="100.0"
        a21="0.0"a22="1.0"a23="0.0"a24="200.0"
        a31="0.0"a32="0.0"a33="1.0"a34="0.0"
    >
        <step:ref_product name="IC1_PACKAGE"/>
        <step:ref_product name="IC2_PACKAGE"/>
    </affine_transformation>
</reference>
```

在下面的示例中,STEP 文件中所有对象的大小都减小到 1/2。

```
<reference xmlns:step="http://www.jeita.or.jp/LPB/step"
    reffile="xxx.stp"format="STEP">
    <affine_transformation
        a11="0.5"a12="0.0"a13="0.0"a14="0.0"
        a21="0.0"a22="0.5"a23="0.0"a24="0.0"
        a31="0.0"a32="0.0"a33="0.5"a34="0.0"
    />
</reference>
```

8.2.12.5.5 sat:ref\_body 语句

8.2.12.5.5.1 概述

此元素用于指定 SAT 文件中通过仿射变换变换坐标系统的对象。放置在指定对象的较低层次上

的对象的坐标系也通过相同的仿射变换矩阵进行变换。通常,此属性指定所引用的 SAT 文件的顶级对象。如果未指定对象,则使用相同的仿射空间变换矩阵对 SAT 文件中的所有对象进行变换。

```
<step:ref_body
    name="objective_name_in_SAT_file "
>
    [<extensions> element]..
</step:ref_body>
```

8.2.12.5.5.2 属性定义

*<sat:ref\_body>* 语句的属性定义如下:  
name: 该属性指定 STEP 文件中对象的名称,该对象的坐标系通过仿射变换进行变换。

8.2.12.5.5.3 例句

在下面的例子中,IC3\_PACKAGE 对象绕 Z 轴旋转 90°,IC4\_PACKAGE 绕 Z 轴旋转 90°。

```
<reference xmlns:sat="http://www.jeita.or.jp/LPB/SAT"
    reffile="xxx.sat" format="sat">
    <affine_transformation
        a11="0.0" a12="1.0" a13="0.0" a14="0.0"
        a21="-1.0" a22="0.0" a23="0.0" a24="0.0"
        a31="0.0" a32="0.0" a33="1.0" a34="0.0"
    >
        <sat:ref_body name="IC3_PACKAGE"/>
    </affine_transformation>
    <affine_transformation
        a11="0.0" a12="-1.0" a13="0.0" a14="0.0"
        a21="1.0" a22="0.0" a23="0.0" a24="0.0"
        a31="0.0" a32="0.0" a33="1.0" a34="0.0"
    >
        <sat:ref_body name="IC4_PACKAGE"/>
    </affine_transformation>
</reference>
```

8.2.12.6 material 语句

8.2.12.6.1 概述

*<material>* 语句用于在引用的几何文件中定义对象的材质。材料的物理参数在 r 格式文件中定义。

```
<material
    name="material_name"
    ref_rule_name="rule_name_in_which_material_is_defined"
>
    [<step:ref_product> element 1
```

```
        <sat:ref_body>element>]....  
        [<extensions>element]...  
    </material>
```

8.2.12.6.2 属性定义

- <material>语句的属性定义如下：
- a) name:此属性定义 R 格式文件中的材料名称；
  - b) ref\_rule\_name:指定定义材料的 R 格式文件中<Physicaldesign>语句的规则名称。

8.2.12.6.3 语句内容

<material>语句包含以下内容。将材料指定给<step:ref\_product>和<sat:ref\_body>语句中定义的对象。如果未指定<step:ref\_product>,则步骤文件中的所有对象都由该材料制成。如果未指定<sat:ref\_body>,则 sat 文件中的所有对象都由该材料制成。

```
    <step:ref_product>  
    <sat:ref_body>  
    <extensions>
```

8.2.12.6.4 step:ref\_product 语句

8.2.12.6.4.1 概述

<step:ref\_product>语句指定参考 STEP 文件中指定材料的对象。指定的材料不会延伸到放置在指定对象的较低层次的对象中。

```
    <step:ref_product  
        name="object_name_in_STEP_file "  
    >  
        [<extensions> element]...  
    </step:ref_product>
```

8.2.12.6.4.2 属性定义

- <step:ref\_product>语句的属性定义如下：
- name:该属性指定 STEP 文件中要指定材料的对象的名称。

8.2.12.6.4.3 例句

在以下示例中,指定 DIE 对象为硅,指定 LEADFRAME01 和 LEADFRAME02 对象为 42Alloy。

```
<reference xmlns:step="http://www.jeita.or.jp/LPB/step"  
    reffile="xxx.stp"format="STEP">  
    <material name="Silicon"ref_rule_name="PartsRule">  
        <step:ref_product name="DIE"/>  
    </material>  
    <material name="42Alloy"ref_rule_name="PartsRule">  
        <step:ref_product name="LEADFRAME01"/>  
        <step:ref_product name="LEADFRAME02"/>
```

```
    </material>
  </reference>
```

8.2.12.6.5 sat:ref\_body 语句

8.2.12.6.5.1 概述

<sat:ref\_body>语句指定 SAT 文件中要指定材料的对象。指定的材料不会延伸到放置在指定对象的较低层次中的对象中。

8.2.12.6.5.2 属性定义

<sat:ref\_body>语句的属性定义如下：  
name:该属性指定 SAT 文件中要指定材料的对象的名称。

8.2.12.6.5.3 例句

以下是<sat:ref\_body>语句的一个示例。

```
<reference xmlns:sat="http://www.jeita.or.jp/LPB/SAT"
           reffile="xxx.sat"format="sat">
  <material name="Silicon"ref_rule_name="PartsRule">
    <sat:ref_body name="die"/>
  </material>
  <material name="42Alloy"ref_rule_name="PartsRule">
    <sat:ref_body name="leadframe01"/>
    <sat:ref_body name="leadframe02"/>
  </material>
</reference>
```

8.2.12.7 heat\_source 语句

8.2.12.7.1 概述

<heat\_source>语句用于定义参考几何文件中的热源。

```
<heat source
  [min="minimum_power "]
  typ="typical_power "
  [max="maximum_power "]
>
  [<step:ref_product>element 1
  <sat:ref_body>element]>....
  [<extensions>element]...
</heat_source>
```

8.2.12.7.2 属性定义

<heat\_source>语句的属性定义如下：



- a) typ:此属性指定典型功率,热值的单位由<unit>语句中的<power>语句定义;
- b) min、max:这些属性指定功率的扰动。最大和最小属性分别是最大和最小功率。功率单位由<unit>语句中的<power>语句定义。

8.2.12.7.3 语句内容

<heat\_source>语句包含以下内容:  
<step:ref\_product>  
<sat:ref\_body>  
<extensions>  
<step:ref\_product>语句和<sat:ref\_body>语句用于定义参考文件中的热源对象。  
如果未定义任何对象,则热量是由参照文件中的整个对象产生的。

8.2.12.7.4 step:ref\_product 语句

8.2.12.7.4.1 概述

<step:ref\_product>语句在引用的 STEP 中指定一个对象以指定该热值。热量是由整个组产生的,包括放置在指定对象的较低层次中的对象。

```
<step:ref_product
    name="object_name_in_STEP_file "
>
    [<extensions> element]...
</step:ref_product>
```

8.2.12.7.4.2 属性定义

name:该属性指定 STEP 文件中要指定热量值的对象的名称。

8.2.12.7.4.3 例句

以下是<step:ref\_product>语句的一个示例。

```
<reference xmlns:step="http://www.jeita.or.jp/LPB/step"
    reffile="xxx.stp"format="STEP">
    <heat_source typ="5.0">
        <step:ref_product name="die_soc"/>
    </heat_source>
    <heat_source typ="0.2">
        <step:ref_product name="die_ddr0"/>
        <step:ref_product name="die_ddr1"/>
    </heat_source></reference>
```

8.2.12.7.5 sat:ref\_body 语句

8.2.12.7.5.1 概述

<sat:ref\_body>语句指定引用 SAT 文件中的一个对象以指定该热值。热量是由整个组产生的,包

括放置在指定对象的较低层次中的对象。

```
<step:ref_body
    name="object_name_in_SAT_file "
>
    [<extensions> element]...
</step:ref_body>
```

8.2.12.7.5.2 属性定义

<sat:ref\_body>语句的属性定义如下：  
name:该属性指定 SAT 文件中要指定热量值的对象的名称。

8.2.12.7.5.3 例句

以下是<sat:ref\_body>语句的一个示例。

```
<reference xmlns:sat="http://www.jeita.or.jp/LPB/sat"
    reffile="xxx.sat" format="SAT">
    <heat_source typ="5.0">
        <sat:ref_body name="die_soc"/>
    </heat_source>
    <heat_source typ="0.2">
        <sat:ref_body name="die_ddr0"/>
        <sat:ref_body name="die_ddr1"/>
    </heat_source>
</reference>
```

8.3 component 语句

8.3.1 概述

<component>语句作为<LPB\_CFORMAT>的直接子语句,此处仅出于兼容性目的进行定义。较新的应用程序应使用<module>语句下的<component>语句。  
<component>语句示例化组件提供物理信息,例如布局。

```
<component>
    [<placement> element]...
    [<extensions> element]...
</component>
```

8.3.2 语句组成

<component>语句包含以下内容：  
<placement>  
<extensions>

8.3.3 placement 语句

8.3.3.1 概述

⟨placement⟩语句定义了如何放置组件。

```
⟨placement
    ref_module="name_of_referenced_module "
    inst="instance_name "
    [distance="unit_of_length "]
    [angleunit="unit_of_angle "]
    [scale="geometrical_scale "]
    [x="x_coordinate "
    y="y_coordinate "]
    [z="z_coordinate "]
    [flip="flip_type "]
    [angle="rotation_angle "]
    [mount="mount_type "]
>
    [⟨extensions⟩ element]...
/⟩
```

8.3.3.2 定义属性

⟨placement⟩语句的属性定义如下。

- a) ref\_module:该属性通过引用模块的属性来指定要放置的模块。
- b) inst:子模块的实例名称。实例名称在父模块中应是唯一的。
- c) distance:此属性定义 x/y 坐标的单位,如果未指定此属性,则使用在⟨unit⟩语句中的⟨distance⟩语句定义的单元,该值应为下列之一:
  - pm:皮米;
  - nm:纳米;
  - μm:微米;
  - mm:毫米;
  - m:米。
- d) angleunit:此属性定义角度的单位,如果未指定此属性,则使用在⟨unit⟩语句中的⟨angle⟩语句处定义的单元,参数应为下列任何一项:
  - degree;
  - radian。
- e) scale:此属性定义距离的比例率,比例率应大于零,如果未指定此属性,则使用 1.0 作为默认属性。
- f) x/y:这些属性定义组件的参考点的位置,x 和 y 属性分别指定 x 坐标和 y 坐标。坐标的单位由⟨unit⟩语句中的⟨distance⟩语句定义。
- g) z:此属性定义组件的 z 坐标,即组件从顶层起始的高度,如图 49 所示,如果未指定此属性,则将零设置为默认值。坐标的单位由⟨unit⟩语句中的⟨distance⟩语句定义。

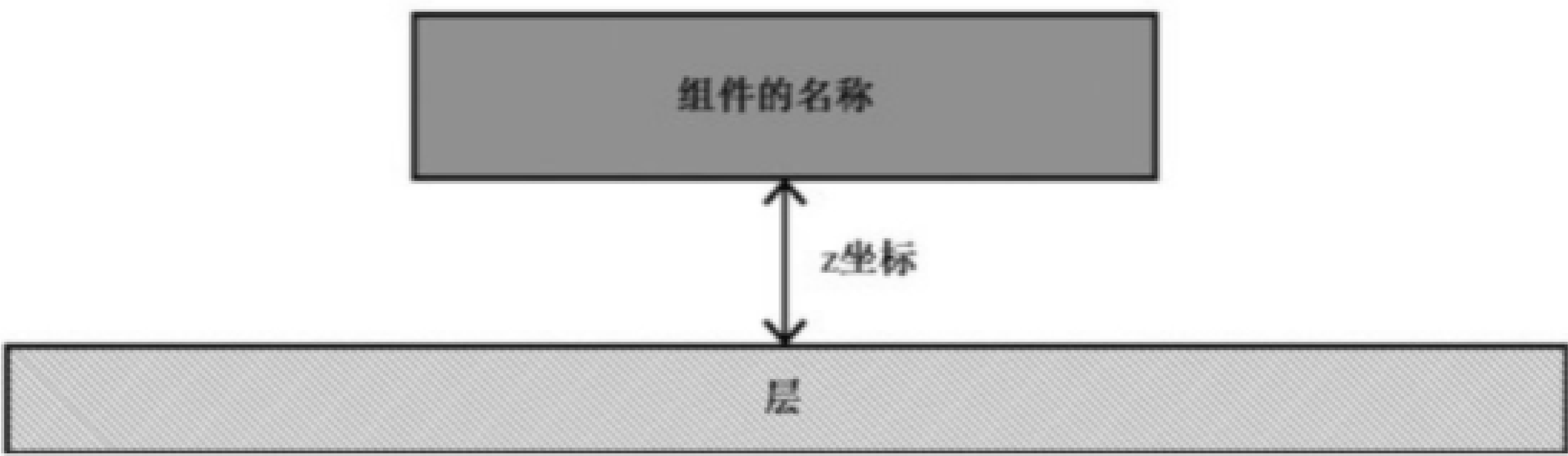


图 49 组件 z 坐标示意图

h) flip:此属性定义倒装的类型,参数应为下列任何一项:

- X 在 X 轴上倒装
- Y 在 Y 轴上倒装

图 50 显示了 X 方向倒装和 Y 方向倒装例子。

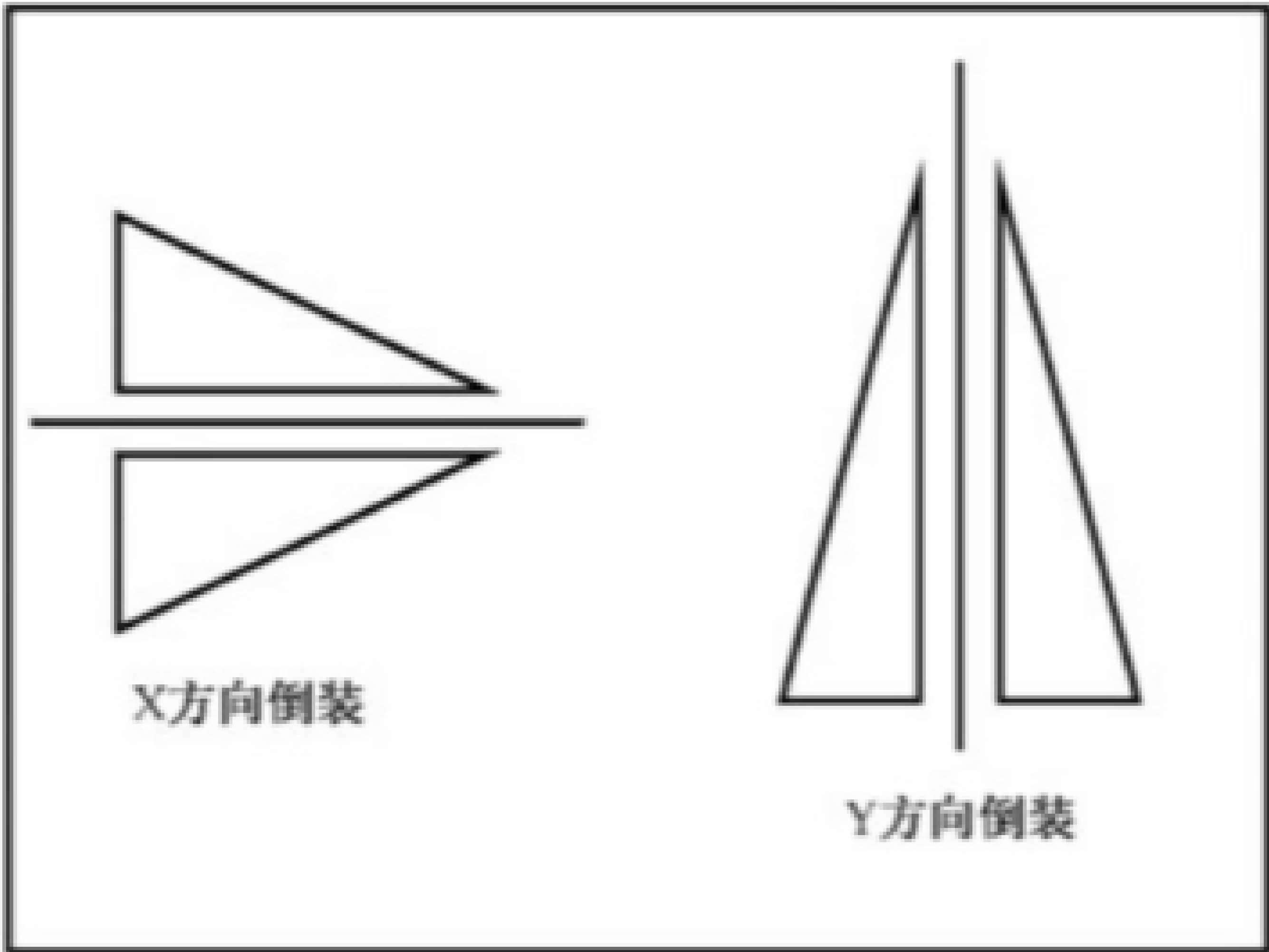


图 50 X 方向倒装和 Y 方向倒装示意图

- i) angle:该属性定义逆时针旋转相对于组件的参考点的角度,如果没有指定角度,则将零设置为默认值。旋转角度的单位由<unit>语句中的<angle>语句定义。
- j) mount:该属性指定模块的位置。当未指定安装层时,见图 51,该值应为以下值之一:
  - TOP 组件放在基板的顶部;
  - BOTTOM 组件放在基板的底部;
  - MIDDLE 模块嵌入基板的内部。

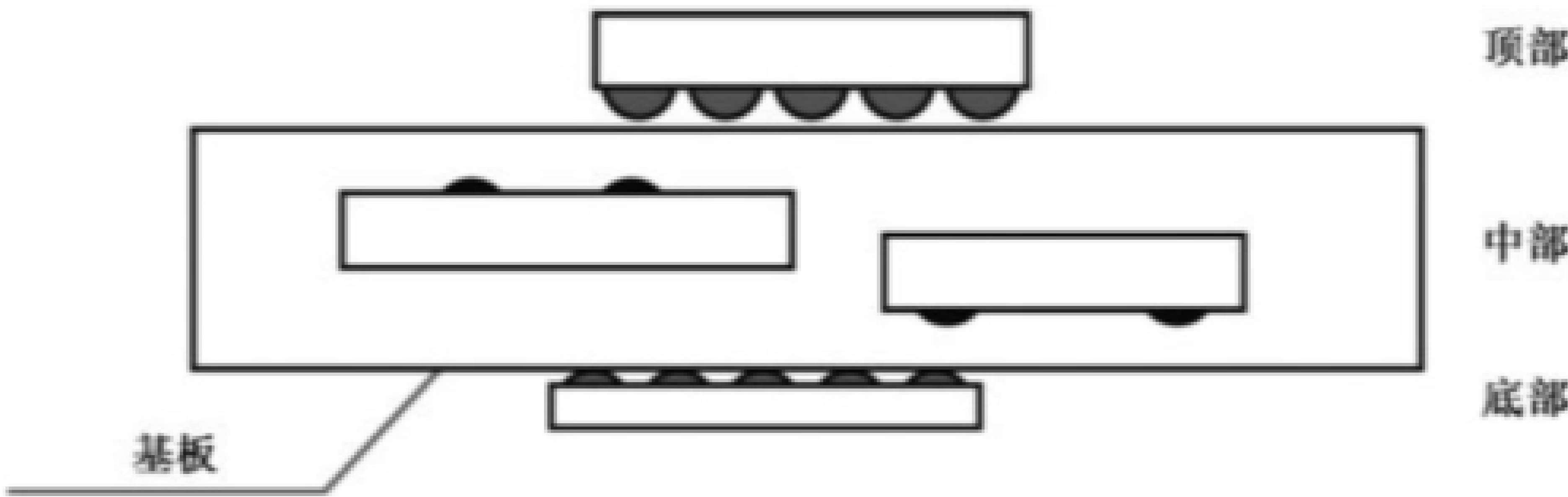


图 51 顶部和底部组件安装示意图

当指定了安装层时,见图 52,该值应为以下值之一:

- TOP 模块放置在安装层上方;
- BOTTOM 模块位于安装层下方;
- MIDDLE 模块放置在安装层内部。

当该属性被允许时,假定为 TOP。

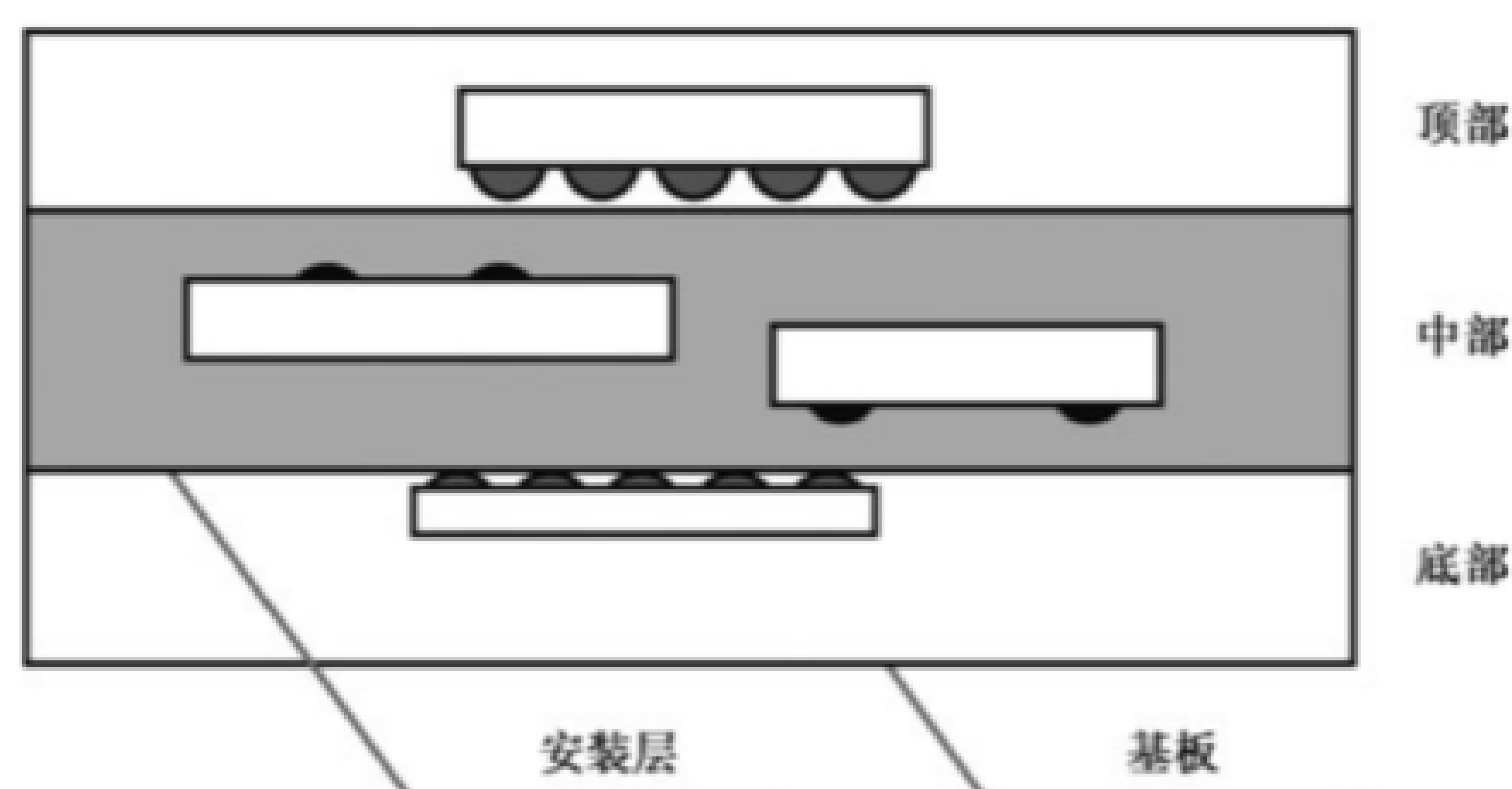


图 52 规定安装层时的组件安装示意图

### 8.3.3.3 例句

以下是〈placement〉语句的一个示例。在下面的例子中,五个单元被放置在 PCB 的顶部,六个单元被放在 PCB 的底部(参见图 53)。

〈component〉

```

    〈placement ref_module="CAP0603B" inst="C10"
      x="-8584.7" y="-4104.9" mount="BOTTOM" />
    〈placement ref_module="CAP0603B" inst="C11"
      x="-8584.7" y="-6355.9" mount="BOTTOM" />
    〈placement ref_module="CAP01005B" inst="C12"
      x="26092.5" y="37686.9" angle="90" mount="BOTTOM" />
    〈placement ref_module="CAP1005B" inst="C13"
      x="30005.4" y="37686.8" angle="90" mount="BOTTOM" />
    〈placement ref_module="CAP1005B" inst="C14"
      x="34659.6" y="37686.8" angle="90" mount="BOTTOM" />
    〈placement ref_module="CAP1005B" inst="C15"
      x="40178.8" y="37686.8" angle="90" mount="BOTTOM" />

    〈placement ref_module="RAS8" inst="RAS8_RN2"
      x="25477.7" y="15729.6" angle="270" mount="TOP" />
    〈placement ref_module="RAS8" inst="RAS8_RN4"
      x="43929.3" y="6225.3" angle="270" mount="TOP" />
    〈placement ref_module="REGULATOR" inst="REGULATOR"
      x="-12598" y="10183.9" mount="TOP" />
    〈placement ref_module="SOC_PKG" inst="SOC"
      x="400" y="-6500" mount="TOP" />
    〈placement ref_module="XTAL" inst="XTAL"
      x="-6285.9" y="26473" mount="TOP" />

```

〈/component〉

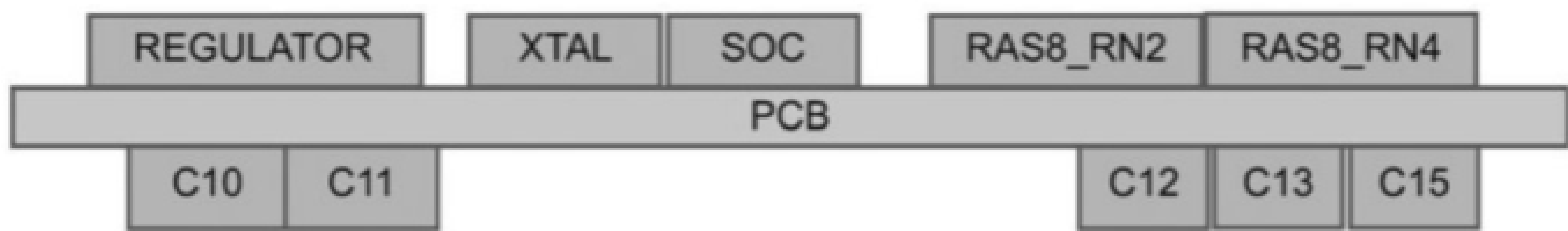


图 53 芯片放置示例

以下是芯片放置的另一个示例。在本例中,两个芯片(Digital\_IC,pll\_cp)堆叠在一个芯片(Interposer\_Die)的顶部(参见图 54)。

```
<component>
  <placement ref_module="DIE" inst="Interposer_Die" x="0" y="0" stack="1">
    <placement ref_module="PLL" inst="PLL_CP " x="15" y="-5" stack="2">
      <placement ref_module="IC" inst="DigitalIC " x="-25" y="0" stack="2">
    </ component >
```

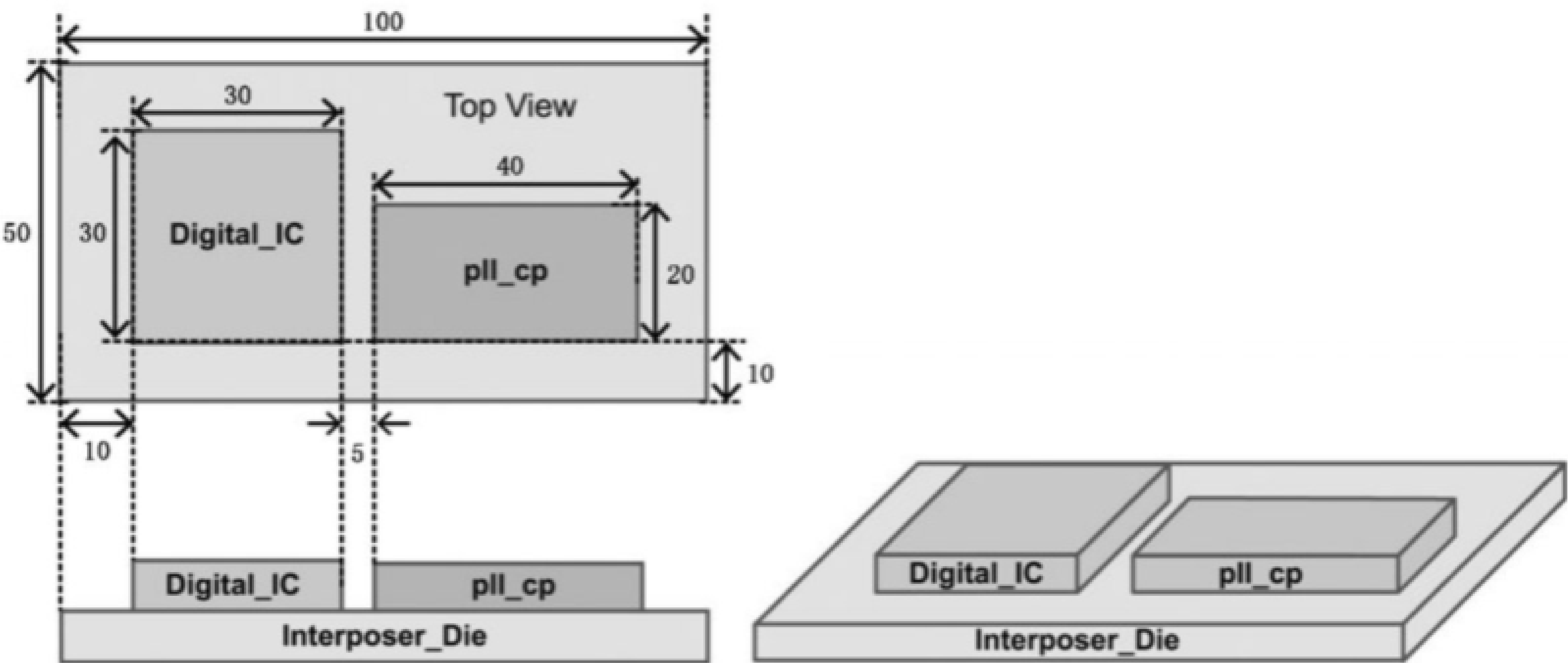


图 54 芯片堆叠放置示例

9 R 格式

9.1 R 格式文件结构

R 格式文件的内容包括一个<header>语句、一个<global>语句、至少一个<Physicaldesign>语句,以及零个或一个<Constraintrule>语句。这些语句应符合以下顺序:

```
<LPB_RFORMAT version="2020">
  <header> element
  <global> element
  {<Physicaldesign> element}...
  [<Constraintrule> element]
  [<extensions> element]...
</LPB_RFORMAT>
```

9.2 Physicaldesign 语句

9.2.1 概述

〈Physicaldesign〉语句定义了物理参数,如导体和介质的材料、叠层、设计规则、通孔、键合丝、球和模具。

```
〈Physicaldesign
    name="design_rule_name "
〉

    [〈default/〉]
    [〈material_def〉 element]
    [〈layer_def〉 element]
    [〈spacing_def〉 element]
    [〈pitch_def〉 element]
    [〈bondingwire_def〉 element]
    [〈ball_def〉 element]
    [〈mold〉 element]
    [〈condctor_struct〉 element]
    [ 〈extensions〉 element ]...
〈/Physicaldesign〉
```

〈Physicaldesign〉语句由名称属性、零或一个〈default〉语句、零或一个〈material\_def〉语句、零或一个〈layer\_def〉语句、零或一个〈spacing\_def〉语句、零或一个〈pitch\_def〉语句、零或一个〈bondingwire\_def〉语句、零或一个〈ball\_def〉语句、零或一个〈mold〉语句、零或一个〈conductor\_struct〉语句组成。

9.2.2 定义属性

〈Physicaldesign〉语句的属性定义如下：  
name:此属性定义设计规则的名称。

9.2.3 语句组成

〈Physicaldesign〉语句包含以下内容：

- 〈default/〉
- 〈material\_def〉
- 〈layer\_def〉
- 〈spacing\_def〉
- 〈pitch\_def〉
- 〈bondingwire\_def〉
- 〈ball\_def〉
- 〈mold〉
- 〈condctor\_struct〉
- 〈extensions〉

## 9.2.4 例句

以下是〈Physicaldesign〉语句的示例。

```

〈Physicaldesign name="default_rule"〉
  〈default/〉
  〈material_def〉
    〈conductor
      material="COPPER"
      volume_resistivity="1.68e-8"
      temperature="20"
    /〉
  /〉
  〈/material_def〉
  〈layer_def〉
    〈layer name="L1"
      type="conductor"
      thickness="10"
      conductor_material="COPPER"〉
      〈line_width min="40"/〉
    /〉
  /〉
  〈/layer_def〉
  〈spacing_def〉
    〈layer name="L1"〉
      〈line_to_line space="40"/〉
      〈line_to_via via="VIA_L1_L2" space="40"/〉
      〈line_to_polygon space="55"/〉
    /〉
  /〉
  〈/spacing_def〉
  〈bondingwire_def〉
    〈bondingwire name="WIREBOND1" diameter="20" material="GOLD"〉
      〈forward horizontal_length="0" vertical_length="100"/〉
      〈forward vertical_length="0" horizontal_ratio="0.125"/〉
      〈length min="500" max="3000"/〉
    /〉
  /〉
  〈/bondingwire_def〉
  〈ball_def〉
    〈ball name="BGA_Ball" material="SOLDER"〉
      〈frustum height="250" diam1="300" diam2="300"/〉
    /〉
  /〉
  〈/ball_def〉
  〈mold width="12000" depth="12000" height="600" material="RESIN" /〉
  〈conductor_struct〉
    〈trapezodial_angle layer="L1" angle="60"/〉

```



```
        <surface_roughness layer="L1" UP_RMS="2" DOWN_RMS="5"/>
    </conductor_struct>
</Physicaldesign>
```

9.2.5 default 语句

<default>语句定义<Physicaldesign>语句作为整个区域的默认设计规则。

```
<default/>
```

9.2.6 material\_def 语句

9.2.6.1 概述

<material\_def>语句定义导体和介质的材料。

```
<material_def>
    [<conductor> element]...
    [<dielectric> element]...
    [<extensions> element ]...
</material_def>
```

<material\_def>语句由零或多个 <conductor>语句和零或多个<dielectric>语句组成。

9.2.6.2 语句构成

<material\_def>语句包含以下内容：

```
<conductor>
<dielectric>
<extensions>
```

9.2.6.3 例句

以下是<material\_def>语句的一个示例。

```
<material_def>
<conductor
    material="COPPER"
    thermal_conductivity="398"
    emissivity="0.06"
    specific_heat_capacity="385"
    density="8.92e3"
    <temperature_characteristic
        temperature="0"
        volume_resistivity="1.55e-8"/>
    <temperature_characteristic
```

```

    temperature="20"
    volume_resistivity="1.68e-8"/>
  <temperature_characteristic
    temperature="100"
    volume_resistivity="2.23e-8"/>
  <temperature_characteristic
    temperature="300"
    volume_resistivity="3.6e-8"/>
</conductor >
<conductor
  material="GOLD"
  thermal_conductivity="320"
  emissivity="0.02"
  specific_heat_capacity="129"
  density="19.3e3"
  <temperature_characteristic
    temperature="0"
    volume_resistivity="2.05e-8"/>
  <temperature_characteristic
    temperature="20"
    volume_resistivity="2.33e-8"/>
  <temperature_characteristic
    temperature="100"
    volume_resistivity="2.88e-8"/>
  <temperature_characteristic
    temperature="300"
    volume_resistivity="4.63e-8"/>
</conductor>
<conductor
  material="SOLDER"
  volume_resistivity="2.17e-7"
  temperature="20"/>
  thermal_conductivity="49"
  emissivity="0.35"
  specific_heat_capacity="176"
  density="9.0e3" />
<dielectric
  material="FR-4"
  thermal_conductivity="0.45"
  emissivity="0.91"
  specific_heat_capacity="950"
  density="1800">
  <frequency_characteristic
```

```
        frequency="1e6"  
        permittivity="4.7"  
        tan_delta="0.010"/>  
    <frequency_characteristic  
        frequency="1e9"  
        permittivity="4.3"  
        tan_delta="0.011"/>  
    <frequency_characteristic  
        frequency="1e10"  
        permittivity="4.1"  
        tan_delta="0.012"/>  
</dielectric>  
<dielectric  
    material="RESISTOR_INK"  
    permittivity="4.5"  
    tan_delta="0.035"  
    frequency="1e9"/>  
<dielectric  
    material="REGIN"  
    permittivity="4.5"  
    tan_delta="0.035"  
    frequency="1e9"  
    thermal_conductivity="2"  
    emissivity="0.85"  
    specific_heat_capacity="820"  
    density="1.73e3"/>  
</material_def>
```

9.2.6.4 conductor 语句

9.2.6.4.1 概述

<conductor>语句定义导体的特性。

```
<conductor  
    material="material_name"  
    [volume_resistivity="resistivity"]  
    [temperature="temperature"]  
    [thermal_conductivity="thermal_conductivity"]  
    [emissivity="emissivity"]  
    [specific_heat_capacity="specific_heat_capacity"]  
    [density="density"]  
    [thermal_diffusivity="thermal_diffusivity"]  
    [coefficient_of_thermal_expansion="coefficient_of_thermal_expansion"]
```

```
[dynamic_viscosity="dynamic_viscosity "]
[molecular_weight="molecular_weight "]
>
    [<temperature_characteristic> element]...
    [<extensions> element]...
</conductor >
```

<conductor>语句由 material 属性、可选 volume\_resistivity 属性、可选 temperature 属性、可选 thermal\_conductivity 属性、可选 emissivity 属性、可选 specific\_heat\_capacity 属性、可选 density 属性、可选 thermal\_diffusivity 属性、可选的 coefficient\_of\_thermal\_expansion 属性、可选的 dynamic\_svisity 属性、可选 molecular\_weight 属性以及零个或多个<temperature\_characteric>和<extensions>语句组成。

9.2.6.4.2 定义属性

- <conductor>语句的属性定义如下。
- a) material:此属性定义导体的名称。
  - b) volume\_resistivity:此属性定义导体的体积电阻率。体积电阻率的单位由<unit>语句中的<resistivity>语句定义。
  - c) temperature:此属性定义测量导体体积电阻率时的温度条件。温度的单位由<unit>语句中的<temperature>语句定义。
  - d) thermal\_conductivity:该属性指定导体的热导率。热导率的单位由<unit>语句中的<thermal\_conductor>语句定义。
  - e) emissivity:该属性指定导体的发射率。
  - f) specific\_heat\_capacity:该属性指定导体的比热容。比热容的单位由<unit>语句中的<specific\_heat\_capacity>语句定义。
  - g) density:该属性指定导体的密度。密度的单位由<unit>语句中的<density>语句定义。
  - h) thermal\_diffusivity:该属性指定热扩散率。热扩散率的单位由<unit>语句中的<thermal\_dispersion>语句定义。
  - i) coefficient\_of\_thermal\_expansion:该属性指定热膨胀系数(CTE)。热膨胀系数的单位由<unit>语句中的<coefficient\_of\_thermal\_expression>语句定义。
  - j) dynamic\_viscosity:该属性指定动态黏度。动态黏度的单位由<unit>语句中的<viscosity>语句定义。
  - k) molecular\_weight:该属性指定分子质量或相对分子质量。

9.2.6.4.3 语句构成

<conductor>语句包含以下内容：

```
<temperature_characteristic>
<extensions>
```

9.2.6.4.4 例句

以下是<conductor>语句的一个示例。

```
<conductor
```

```
material="COPPER"
thermal_conductivity ="398"]
emissivity ="0.06 "
specific_heat_capacity ="385 "
density ="8.92e3"
<temperature_characteristic
    temperature="0"
    volume_resistivity="1.55e-8"/>
<temperature_characteristic
    temperature="20"
    volume_resistivity="1.68e-8"/>
<temperature_characteristic
    temperature="100"
    volume_resistivity="2.23e-8"/>
<temperature_characteristic
    temperature="300"
    volume_resistivity="3.6e-8"/>
</conductor >
<conductor
    material="SOLDER"
    volume_resistivity="2.17e-7"
    temperature="20"/>
    thermal_conductivity="49"
    emissivity="0.35"
    specific_heat_capacity="176"
    density="9.0e3"
/>
```

9.2.6.4.5 temperature\_characteristic 语句

9.2.6.4.5.1 概述

<temperature\_characteristic>语句指定导体随温度变化的体积电阻率。

```
<temperature_characteristic
    temperature="temperature "
    volume_resistivity="volume_resistivity "
>
    [<extensions> element]...
</temperature_characteristic>
```

<temperature\_characteristic>语句由 temperature 属性和 volume\_resistivity 电阻率属性组成。

9.2.6.4.5.2 属性定义

<temperature\_characteristic>语句的属性定义如下：

- a) temperature:该属性指定测量导体体积电阻率时的温度条件。温度的单位由<unit>语句中的<temperature>语句定义；
- b) volume\_resistivity:该属性指定导体的体积电阻率。体积电阻率的单位由<unit>语句中的<resistivity>语句定义。

9.2.6.4.5.3 例句

以下是<temperature\_characteristic>语句的一个示例。

```
<temperature_characteristic
    temperature="0"
    volume_resistivity="1.55e-8"
/>>
```

9.2.6.5 Dielectric 语句

9.2.6.5.1 概述

<dielectric>语句定义介质的特性。

```
<dielectric
    material="material_name "
    [permittivity="permittivity "]
    [tan_delta="loss_tangent "]
    [frequency="frequency "]
    [thermal_conductivity =" thermal_conductivity "]
    [emissivity ="emissivity "]
    [specific_heat_capacity ="specific_heat_capacity "]
    [density ="density "]
    [thermal_diffusivity="thermal_diffusivity "]
    [coefficient_of_thermal_expansion="coefficient_of_thermal_expansion "]
    [dynamic_viscosity="dynamic_viscosity "]
    [molecular_weight="molecular_weight "]
>
    [<frequency_characteristic> element]...
    [<extensions> element]...
</dielectric>
```

<dielectric>语句由 material 属性、可选 permittivity 属性、可选 tan\_delta 属性、可选 frequency 属性、可选 thermal\_conductivity 属性、可选 emissivity 属性、可选 specific\_heat\_capacity 属性、可选 density 属性、可选 thermal\_diffusivity 属性,可选的 coefficient\_of\_thermalexpansion 属性、可选的 dynamic\_scredition 属性、任选的 molecular\_weight 属性以及零个或多个 frequency\_characteristic 语句组成。

9.2.6.5.2 定义属性

<dielectric>语句的属性定义如下。

- a) material:此属性定义介电材料的名称。
- b) permittivity:此属性定义材料的介电常数。
- c) tan\_delta:该属性指定材质的耗散因子。
- d) frequency:此属性定义测量介电特性时使用的频率。频率的单位由<unit>语句中的<frequency>语句定义。
- e) thermal\_conductivity:该属性指定导体的热导率。热导率的单位由<unit>语句中的<thermal\_conduction>语句定义。
- f) emissivity:该属性指定导体的发射率。
- g) specific\_heat\_capacity:该属性指定导体的比热容。比热容的单位由<unit>语句中的<specific\_heat\_capacity>语句定义。
- h) density:该属性指定导体的密度。密度的单位由<unit>语句中的<density>语句定义。
- i) thermal\_diffusivity:该属性指定热扩散率。热扩散率的单位由<unit>语句中的<thermal\_dispersions>语句定义。
- j) coefficient\_of\_thermal\_expansion:该属性指定热膨胀系数(CTE)。热膨胀系数的单位由<unit>语句中的<coefficient\_of\_thermal\_expression>语句定义。
- k) dynamic\_viscosity:该属性指定动态黏度。动态黏度的单位由<单位>语句中的<动态黏度>语句定义。
- l) molecular\_weight:该属性指定分子质量或相对分子质量。

9.2.6.5.3 语句内容

<dielectric>语句包含以下内容:

<frequency\_characteristic>  
<extensions>

9.2.6.5.4 例句

以下是<dielectric>语句的一个示例。

```
<dielectric
    material="FR-4"
    permittivity="4.5"
    tan_delta="0.035"
    frequency="1e9"
/>

<dielectric
    material="FR-4"
    thermal_conductivity = "0.45"]
    emissivity = "0.91 "
    specific_heat_capacity = "950"
    density = "1800"
    <frequency_characteristic
        frequency = "1e6"
```

```

                                permittivity="4.7"
                                tan_delta="0.010"  />
    <frequency_characteristic
                                frequency = "1e9"
                                permittivity="4.3"
                                tan_delta="0.011"  />
    <frequency_characteristic
                                frequency = "1e10"
                                permittivity="4.1"
                                tan_delta="0.012  />
</dielectric>
<dielectric
    material="REGIN"
    permittivity="4.5"
    tan_delta="0.035"
    frequency="1e9"
    thermal_conductivity = "2"
    emissivity = "0.85 "
    specific_heat_capacity = "820"
    density = "1.73e3"
/>
```

9.2.6.5.5 frequency\_characteristic 语句

9.2.6.5.5.1 概述

<frequency\_characteristic>语句指定了电介质的介电常数。

```

<frequency_characteristic
    frequency="frequency "
    permittivity="permittivity "
    [tan_delta="tan_delta "]
>
    [<extensions> element]...
</frequency_characteristic>
```

<frequency\_characteristic>语句由频率属性、介电常数属性和可选的 tan\_delta 属性组成。

9.2.6.5.5.2 属性定义

<frequency\_characteristic>语句的属性定义如下：

- a) frequency:此属性指定测量介电特性时使用的频率,频率的单位由<unit>语句中的<frequency>语句定义；
- b) permittivity:该属性指定材料的介电常数；
- c) tan\_delta:该属性指定材料的耗散因子。



9.2.6.5.5.3 例句

以下是<frequency\_characteristic>语句的一个示例。

```
<frequency_characteristic
    frequency="1e6"
    permittivity="4.7"
    tan_delta="0.010"
/>
```

9.2.7 layer\_def 语句

9.2.7.1 概述

<layer\_def>语句定义从顶部到下底部的叠层。

```
<layer_def>
    {<layer> element}...
    [<extensions> element]...
</layer_def>
```

<layer\_def>语句由一个或多个<layer>语句组成。

9.2.7.2 语句构成

<layer\_def>语句包含以下内容：

```
<layer>
<extensions>
```

9.2.7.3 例句

图 55 显示了包括导体层和介电层在内的叠层的一个例子。

```
<layer_def>
    <layer name="TOP_SR"
        type="dielectric"
        thickness="20"
        dielectric_material="RESISTOR_INK"/>
    <layer name="L1"
        type="conductor"
        thickness="10"
        conductor_material="COPPER"
    >
        <line_width min="40"/>
</layer>
```

```

    <layer name="BU_1_2"
      type="dielectric"
      thickness="40"
      dielectric_material="FR-4"
    />
    <layer name="L2"
      type="conductor"
      thickness="10"
      conductor_material="COPPER"
    >
      <line_width min="50"/>
</layer>
<layer name="CORE"
  type="dielectric"
  thickness="100"
  dielectric_material="FR-4"
/>
<layer name="L3"
  type="conductor"
  thickness="10"
  conductor_material="COPPER"
>
  <line_width min="50"/>
</layer>
<layer name="BU_3_4"
  type="dielectric"
  thickness="40"
  dielectric_material="FR-4"
/>
<layer name="L4"
  type="conductor"
  thickness="10"
  conductor_material="COPPER"
>
  <line_width min="40"/>
</layer>
<layer name="BOTTOM_SR"
  type="dielectric"
  thickness="20"
  dielectric_material="RESISTOR_INK"
/>
</layer_def>

```

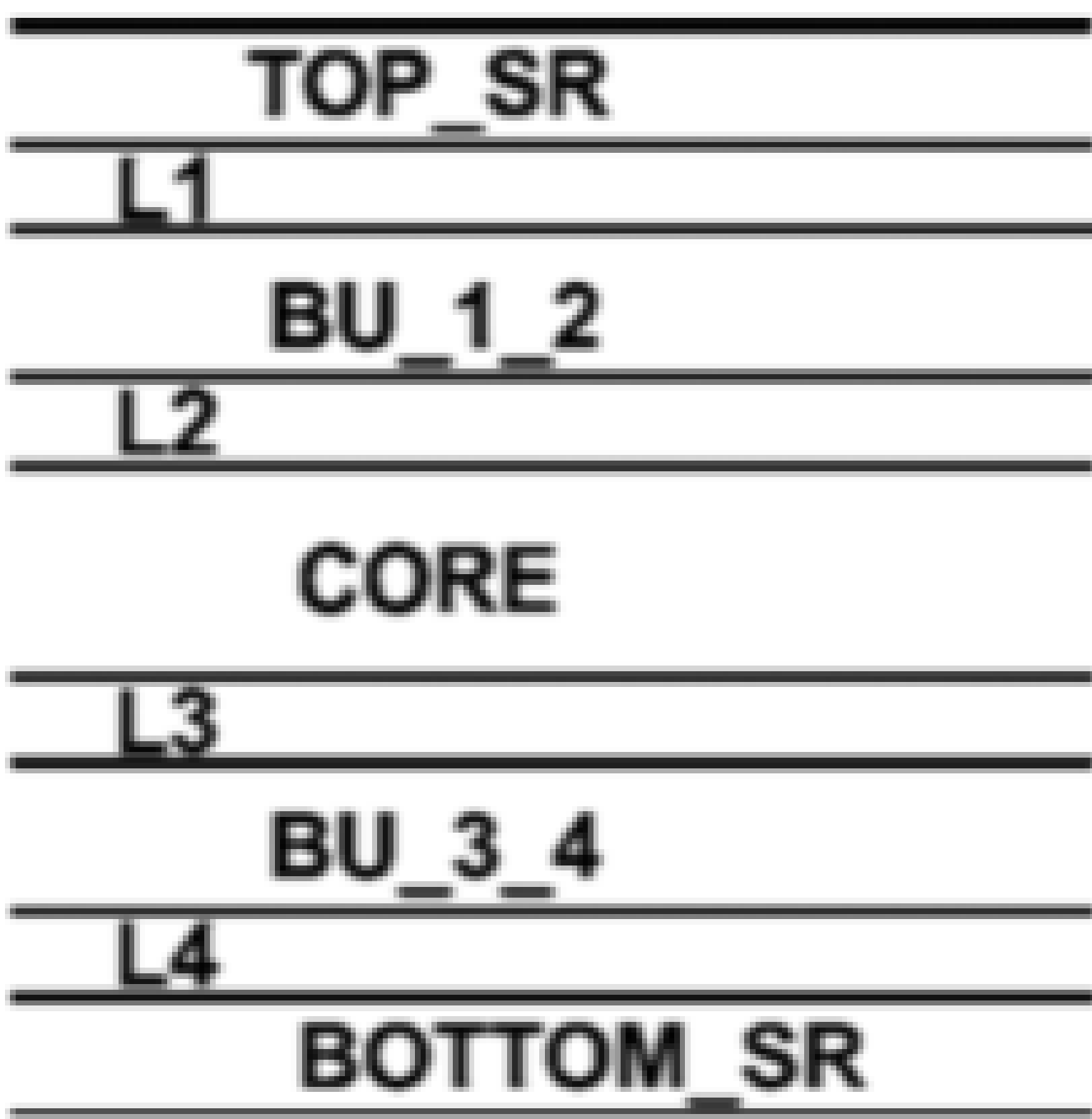


图 55 叠层示意图

9.2.7.4 layer 语句

9.2.7.4.1 概述

⟨layer⟩语句定义层的名称、层的类型、层的厚度、层的材料以及线宽和面积的设计规则。

```
⟨layer
    name="layer_name "
    type="layer_type "
    thickness="layer_thickness "
    [plate_thickness="plating_thickness "]
    [conductor_material="name_of_conductor_material "]
    [dielectric_material="name_of_dielectric_material "]
>
    [⟨line_width⟩ element]
    [⟨area_limit⟩ element]
    [⟨extensions⟩ element]...
⟨/layer⟩
```

⟨layer⟩语句由名称属性、类型属性、厚度属性、可选板厚度属性、可选导体\_材料属性、可选介质\_材料属性、零或一个⟨line\_width⟩语句、零或一个⟨area\_limit⟩语句组成以及零或一个⟨extensions⟩语句组成。

9.2.7.4.2 定义属性

⟨layer⟩语句的属性定义如下。

- a) name:此属性定义层名。
- b) type:此属性定义层的类型。这些参数如下：
  - 1) dielectric 介质层,介质膜;
  - 2) conductor 导体层。
- c) thickness:此属性定义导电层总厚度,即包含电镀层厚度的总厚度,如图 56 所示。厚度的单位由⟨unit⟩语句中的⟨distance⟩语句定义。
- d) plate\_thickness:此属性定义厚度属性中包含的电镀层厚度,如图 56 所示。镀层厚度的单位由⟨unit⟩语句中的⟨distance⟩语句定义。

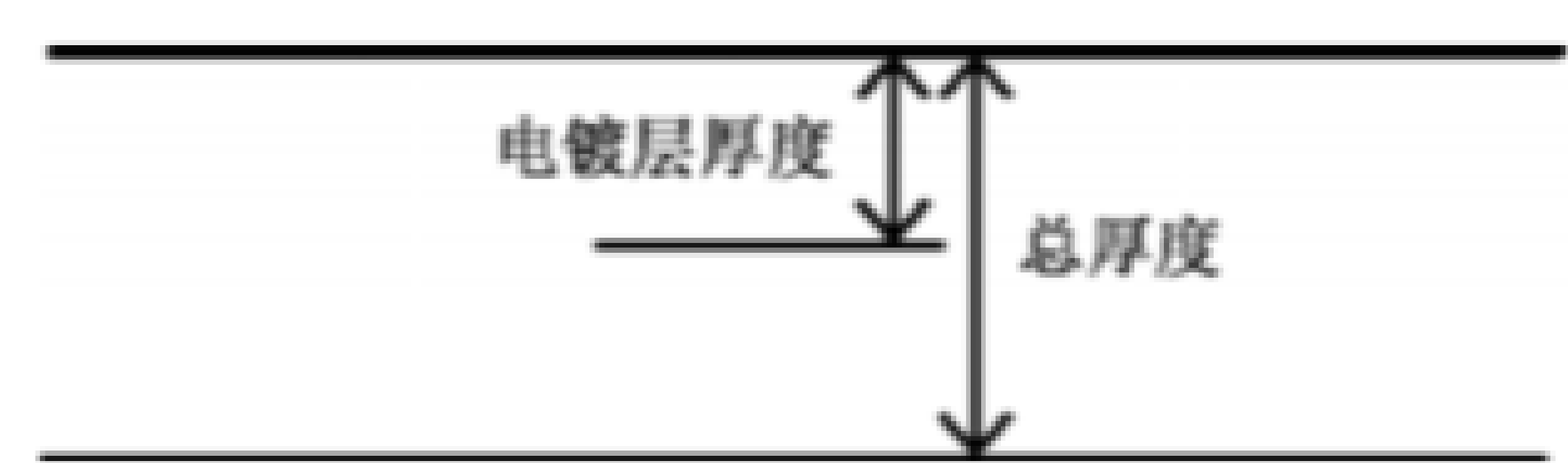


图 56 电镀层厚度与总厚度关系示意图

- e) conductor\_material:此属性定义导体材料。
- f) dielectric\_material:此属性定义介电材料。

9.2.7.4.3 语句构成

⟨layer⟩语句包含以下内容：

⟨line\_width⟩  
⟨area\_limit⟩  
⟨extensions⟩

9.2.7.4.4 例句

以下是⟨layer⟩语句的一个示例。

```
⟨layer
  name="TOP_SR"
  type="dielectric"
  thickness="20"
  dielectric_material="RESISTOR_INK"
/>
⟨layer
  name="L1"
  type="conductor"
  thickness="10"
  conductor_material="COPPER">
  ⟨line_width min="40"/>
</layer>
⟨layer
  name="BU_1_2"
  type="dielectric"
  thickness="40"
  dielectric_material="FR-4"
/>
```

9.2.7.4.5 line\_width 语句

9.2.7.4.5.1 概述

⟨line\_width⟩语句定义线宽的设计规则。

```
<line_width
    min="minimum_line_width "
    [max="maximum_line_width "]
/>
```

<line\_width>语句由 min 属性和可选的 max 属性组成。

9.2.7.4.5.2 定义属性

- <line\_width>语句的属性定义如下：
- a) min:此属性定义最小线宽,线宽的单位由<unit>语句中的<distance>语句定义；
  - b) max:此属性定义最大线宽,线宽的单位由<unit>语句中的<distance>语句定义。

9.2.7.4.5.3 例句

以下是<line\_width>语句的一个示例。

```
<line_width min="40" max="80" />
```

9.2.7.4.6 area\_limit 语句

9.2.7.4.6.1 概述

<area\_limit>语句定义最小面积规则。

```
<area_limit
    min="minimum_area "
/>
```

<area\_limit>语句由 min 属性组成。

9.2.7.4.6.2 2 定义属性

- <area\_limit>语句的属性定义如下：
- min:此属性定义最小金属面积。金属区域的单位由<unit>语句中的<area>语句定义。

9.2.7.4.6.3 例句

以下是<area\_limit>语句的一个示例。

```
<area_limit min="22500"/>
```

9.2.8 spacing\_def 语句

9.2.8.1 概述

<spacing\_def>语句定义了间距的设计规则,例如导线之间的间距、导线与导通孔之间的间距以及直线与多边形之间的间距。

```
<spacing_def>
```

```
        {<layer> element}...
        [<extensions> element]...
</ spacing_def >
```

<spacing\_def>语句由一个或多个<Layer>和<extensions>语句组成。

9.2.8.2 语句构成

<spacing\_def>语句包含以下内容：

```
<layer>
<extensions>
```

9.2.8.3 例句

以下是<spacing\_def>语句的一个示例。

```
<spacing_def>
  <layer name="L1">
    <line_to_line space="40"/>
    <line_to_via via="VIA_L1_L2" space="40"/>
    <line_to_polygon space="55"/>
  </layer>
  <layer name="L2">
    <line_to_line space="50"/>
    <line_to_via space="50"/>
    <line_to_via space="50"/>
    <line_to_polygon space="55"/>
  </layer>
  <layer name="L3">
    <line_to_line space="50"/>
    <line_to_via space="50"/>
    <line_to_via via="VIA_L3_L4" space="50"/>
    <line_to_polygon space="55"/>
  </layer>
  <layer name="L4">
    <line_to_line space="40"/>
    <line_to_via space="40"/>
    <line_to_polygon space="55"/>
  </layer>
</spacing_def>
```

9.2.8.4 layer 语句

9.2.8.4.1 概述

<layer>语句定义了间距的设计规则，例如线之间的间距、线与通孔之间的间距以及线与多边形之

间的间距(参见图 57)。

```
<layer
    name="layer_name "
>
    [<line_to_line> element]
    [<line_to_via> element]...
    [<line_to_polygon> element]
    [<extensions> element]...
</layer>
```

<layer>语句由 name 属性、零或一个<line\_to\_line>语句、零或多个<line\_to\_via>语句和零或一个<line\_to\_polygon>和<extensions>语句组成。

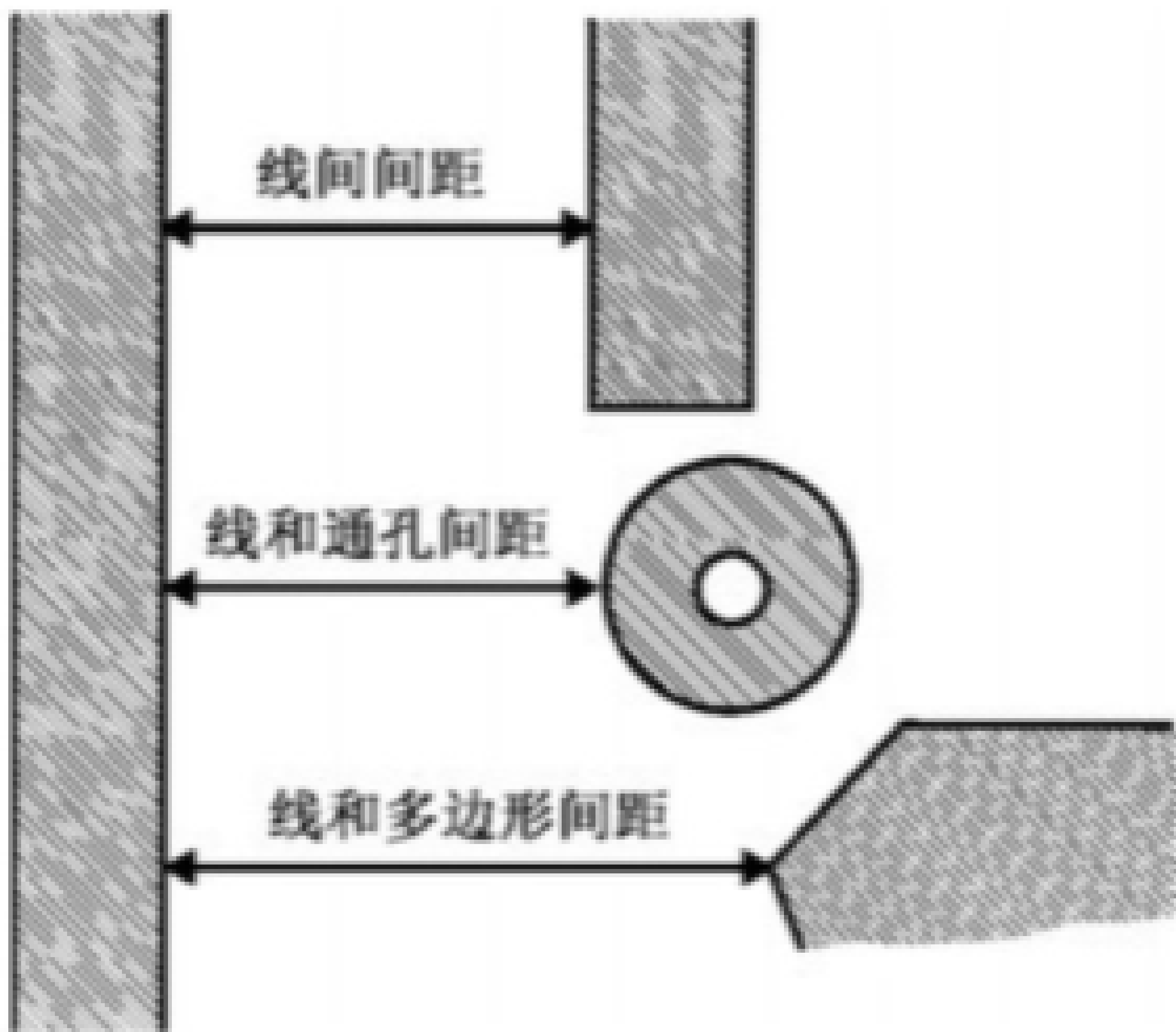


图 57 间距规则示例

9.2.8.4.2 定义属性

<layer>语句的属性定义如下：  
name:此属性定义层的名称。

9.2.8.4.3 语句构成

<layer>语句包含以下内容：

```
<line_to_line>
<line_to_via>
<line_to_polygon>
<extensions>
```

9.2.8.4.4 例句

以下是<layer>语句的一个示例。

```
<layer name="L1">
    <line_to_line space="40"/>
    <line_to_via via="VIA_L1_L2" space="40"/>
```

```
        <line_to_polygon space="55"/>
    </layer>
    <layer name="L2">
        <line_to_line space="50"/>
        <line_to_via space="50"/>
        <line_to_polygon space="55"/>
    </layer>
```

9.2.8.4.5 line\_to\_line 语句

9.2.8.4.5.1 概述

<line\_to\_line>语句定义导线之间的间距。

```
<line_to_line
    space="minimum_spacing "
/>
```

<line\_to\_line>语句由 space 属性组成。

9.2.8.4.5.2 定义属性

<line\_to\_line>语句的属性定义如下：

space：此属性定义导线之间的最小间距。

9.2.8.4.5.3 例句

以下是<line\_to\_line>语句的一个示例。

```
<line_to_line space="50"/>
```

9.2.8.4.6 line\_to\_via 语句

9.2.8.4.6.1 概述

<line\_to\_via>语句定义线和通孔之间的距离。

```
<line_to_via
    [via="identifier_of_padstack "]
    pace="minimun_spacing "
/>
```

<line\_to\_via>语句由可选的 via 属性和 space 属性组成。

9.2.8.4.6.2 定义属性

<line\_to\_via>语句的属性定义如下：

- a) via：此属性定义应用于通孔的焊盘标识符间距规则；
- b) space：此属性定义线和通孔之间的最小间距。间距的单位由<unit>语句中的<distance>语句



定义。

9.2.8.4.6.3 例句

以下是<line\_to\_via>语句的一个示例。

```
<line_to_via via="VIA_L1_L2" space="40"/>
```

9.2.8.4.7 line\_to\_polygon 语句

9.2.8.4.7.1 概述

<line\_to\_polygon>语句定义线和多边形之间的距离。

```
<line_to_polygon
    space="minimum_spacing "
/>
```

<line\_to\_polygon>语句由 space 属性组成。

9.2.8.4.7.2 定义属性

<line\_to\_polygon>语句的属性定义如下：  
space:此属性定义线和多边形最小间距。

9.2.8.4.7.3 例句

以下是<line\_to\_polygon>语句的一个示例。

```
<line_to_polygon space="55"/>
```

9.2.9 pitch\_def 语句

9.2.9.1 概述

<pitch\_def>语句定义从通孔中心到通孔中心的间距。

```
<pitch_def>
    {<via_pitch> element}...
    [<extensions> element]...
</pitch_def>
```

<pitch\_def>语句由一个或多个 <via\_pitch>语句组成。

9.2.9.2 语句构成

<pitch\_def>语句包含以下内容：

```
<via_pitch>
<extensions>
```

9.2.9.3 例句

以下是不能使用叠层通孔的示例。

```
<pitch_def>
  <via_pitch via1="VIA_L1_L2" via2="VIA_L1_L2" pitch="250"/>
  <via_pitch via1="VIA_L1_L2" via2="VIA_L2_L3" pitch="275"
samenet_pitch="225"/>
  <via_pitch via1="VIA_L2_L3" via2="VIA_L2_L3" pitch="300"/>
  <via_pitch via1="VIA_L2_L3" via2="VIA_L3_L4" pitch="275"
samenet_pitch="225"/>
  <via_pitch via1="VIA_L3_L4" via2="VIA_L3_L4" pitch="250"/>
</pitch_def>
```

以下是可以使用叠层通孔的示例：

```
<pitch_def>
  <via_pitch via1="VIA_L1_L2" via2="VIA_L1_L2" pitch="250"/>
  <via_pitch via1="VIA_L1_L2" via2="VIA_L2_L3" pitch="275"
stacked_offset="75"/>
  <via_pitch via1="VIA_L2_L3" via2="VIA_L2_L3" pitch="300"/>
  <via_pitch via1="VIA_L2_L3" via2="VIA_L3_L4" pitch="275"
stacked_offset="75"/>
  <via_pitch via1="VIA_L3_L4" via2="VIA_L3_L4" pitch="250"/>
</pitch_def>
```

9.2.9.4 via\_pitch 语句

9.2.9.4.1 概述

图 58 显示了通孔结构横截面视图的一个例子。<via\_pitch>语句定义了通孔节距的设计规则，如图 59 所示。

```
<via_pitch
  via1="identifier_of_padstack "
  via2="identifier_of_padstack "
  pitch="minimum_pitch "
  [samenet_pitch="minimum_pitch_for_same_net "]
  [stacked_offset="acceptable_gap "]
/>
```

<via\_pitch>语句定义从中心到中心的通孔之间的节距(参见图 59)。<via\_pitch>语句由 via1 属性、via2 属性、pitch 属性、可选 samenet\_pitch 属性和可选 stacked\_offset 属性组成。

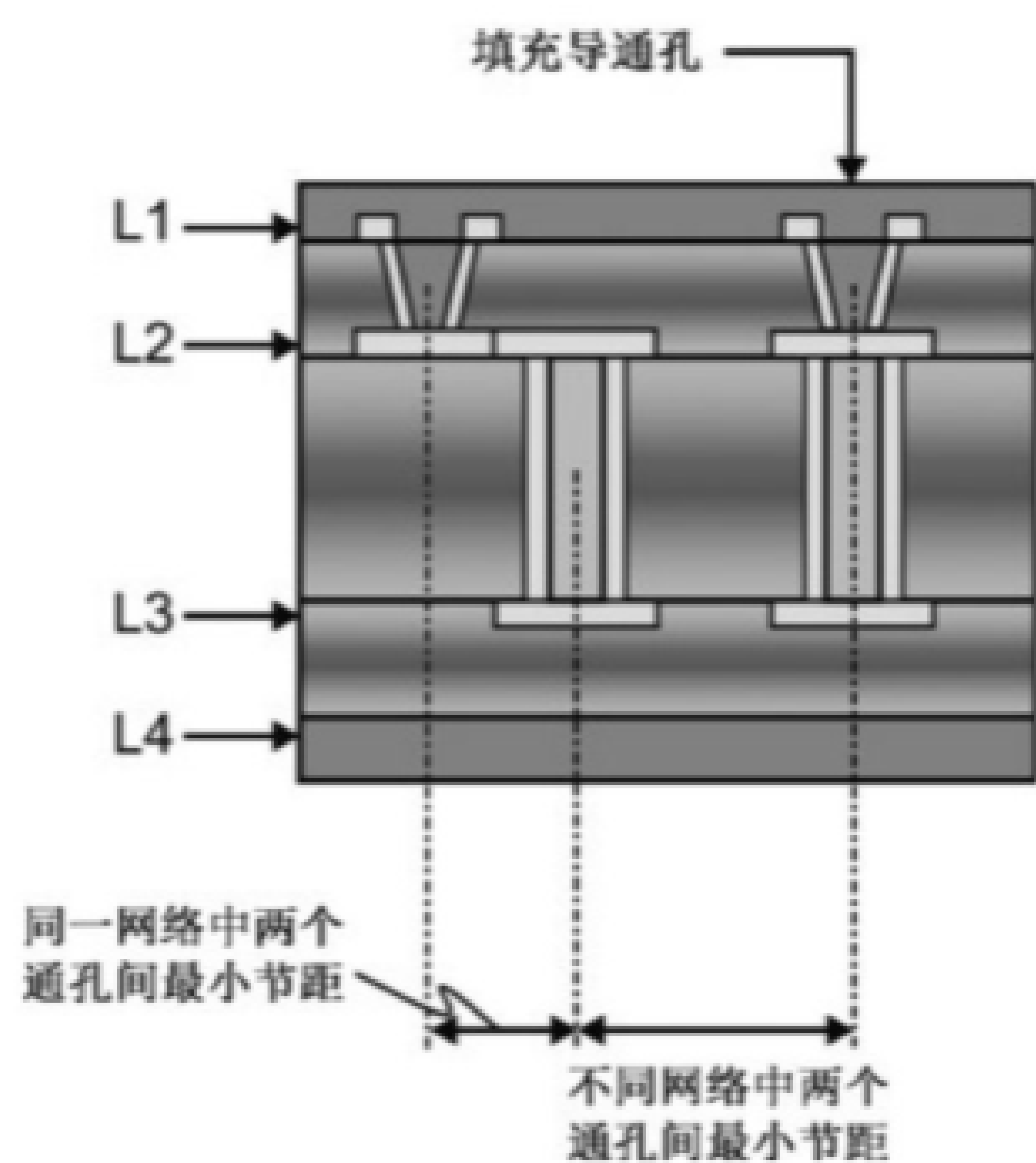


图 58 通孔结构示例



图 59 通孔间最小节距的例子

9.2.9.4.2 定义属性

- 〈via\_pitch〉语句的属性定义如下。
- a) via1:此属性定义应用通孔间距规则的通孔的焊盘标识符,如图 59 所示。
  - b) via2:此属性定义应用通孔间距规则的通孔的焊盘标识符,如图 59 所示。
  - c) pitch:此属性定义不同网络中两个通孔之间的最小节距,如图 59 所示。间距的单位由〈unit〉语句中的〈distance〉语句定义。
  - d) samenet\_pitch:此属性定义同一网络中两个通孔之间的最小节距,如图 59 所示。间距的单位由〈unit〉语句中的〈distance〉语句定义。
  - e) stacked\_offset:此属性定义堆叠导通孔中心之间的最大可接受节距,如图 59 所示。偏移的单位由〈unit〉语句中的〈distance〉语句定义。

9.2.9.4.3 例句

以下是〈via\_pitch〉语句的一个示例。

```
<via_pitch
  via1 = "VIA_L1_L2"
  via2 = "VIA_L1_L2"
  pitch = "250"
/>
<via_pitch
```

```
        via1="VIA_L1_L2"  
        via2="VIA_L2_L3"  
        pitch="275"  
        samenet_pitch="225"  
    />  
    <via_pitch  
        via1="VIA_L1_L2"  
        via2="VIA_L2_L3"  
        pitch="275"  
        stacked_offset="75"  
    />
```

9.2.10 bondingwire\_def 语句

9.2.10.1 概述

<bondingwire\_def>语句定义键合丝的形状和材料。

```
<bondingwire_def>  
    {<bondingwire> element}...  
    [<extensions> element]...  
</bondingwire_def>
```

<bondingwire\_def>语句由一个或多个<bondingwire>语句组成。

9.2.10.2 语句构成

<bondingwire\_def>语句包含以下内容：

```
<bondingwire>  
<extensions>
```

9.2.10.3 例句

以下是<bondingwire\_def>语句的一个示例。

```
<bondingwire_def>  
    <bondingwire name="WIREBOND1" diameter="20" material="GOLD">  
        <forward horizontal_length="0" vertical_length="100"/>  
        <forward vertical_length="0" horizontal_ratio="0.125"/>  
        <length min="500" max="3000"/>  
    </bondingwire>  
</bondingwire_def>
```

9.2.10.4 bondingwire 语句

9.2.10.4.1 概述

<bondingwire>语句定义键合丝的形状和材料。

```
<bondingwire
    name="wire_name "
    diameter="wire_diameter "
    [material="material_name "]
>
    [<forward> element]...
    [<backward> element]...
    [<length> element]
    [<extensions> element]...
</bondingwire>
```

<bondingwire>语句由 name 属性、diameter 属性、可选的 material 属性、零或多个<forward>语句、0 或多个<backward>及<extensions>语句和 0 或 1 个<length>语句组成。

9.2.10.4.2 定义属性

- <bondingwire>语句的属性定义如下：
- a) name:此属性定义键合丝的名称；
  - b) diameter:此属性定义键合丝的直径,直径的单位由<unit>语句中的<distance>语句定义；
  - c) material:此属性定义键合丝的材料。

9.2.10.4.3 语句构成

<bondingwire>语句包含以下内容：

```
<forward>
<backward>
<length>
<extensions>
```

9.2.10.4.4 例句

图 60 显示了由相应的<bondingwire\_def>描述定义的四点键合形状的示例,如下所示：

四点键合：

```
<bondingwire_def>
    <bondingwire name="WIREBOND1" diameter="20" material="GOLD">
        <forward horizontal_length="0" vertical_length="100"/>
        <forward vertical_length="0" horizontal_ratio="0.125"/>
        <length min="500" max="3000"/>
    </bondingwire>
</bondingwire_def>
```

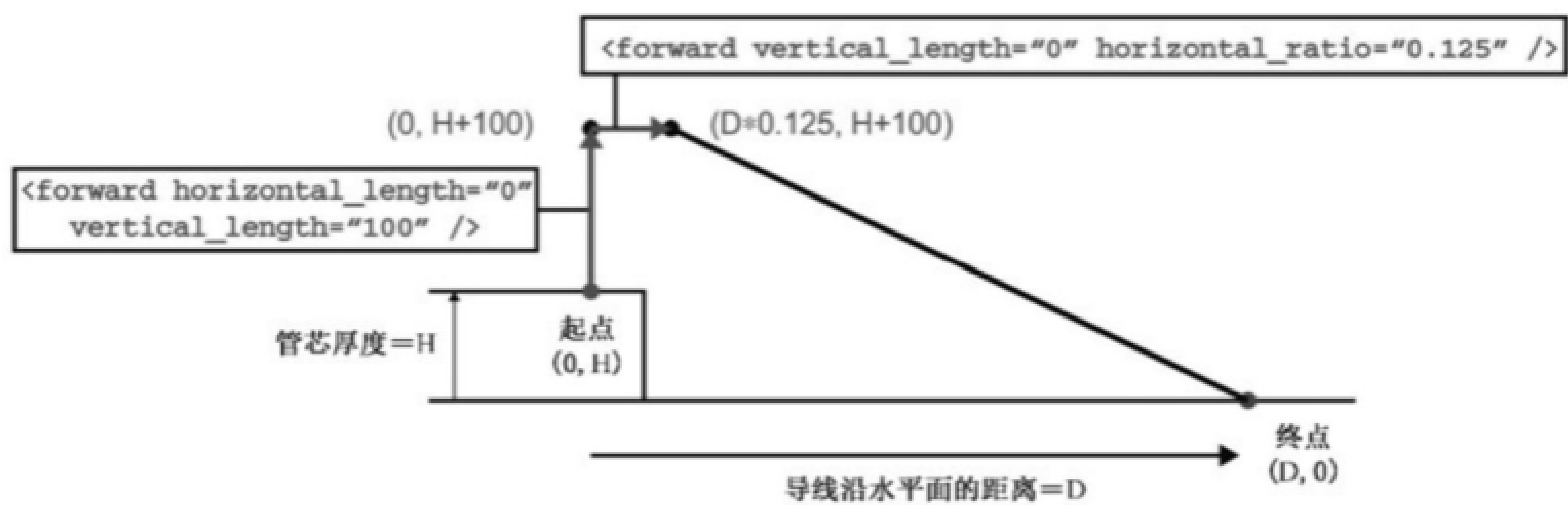


图 60 键合丝形状示例(四点键合)

图 61 显示了由相应的<bondingwire\_def>描述定义的五点键合形状的示例,如下所示:

五点键合:

```
<bondingwire_def>
  <bondingwire name="WIREBOND2" diameter="20" material="GOLD">
    <forward vertical_length="100" angle="80"/>
    <forward vertical_length="0" horizontal_ratio="0.125"/>
    <backward angle="20" horizontal_ratio="0.5"/>
    <length min="500" max="3000"/>
  </bondingwire>
</bondingwire_def>
```

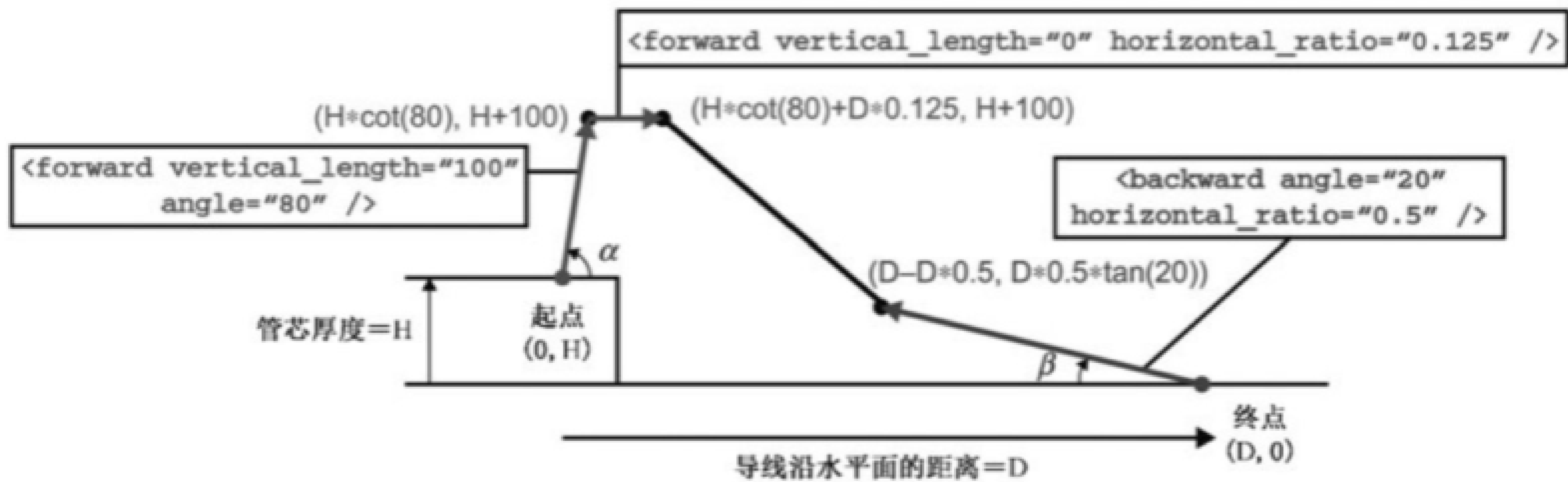


图 61 键合丝形状示例(五点键合)

9.2.10.4.5 forward 语句

9.2.10.4.5.1 概述

<forward>语句定义来自管芯一侧的循环键合丝。

```
<forward
  horizontal_length="horizontal_length "
  vertical_length="vertical_length "
  angle="die_side_angle "
  horizontal_ratio="horizontal_ratio "
```

/>

一些<forward>语句定义从第一个位置到第二个位置趋向键合区(参见图 62)。<forward>语句由可选的水平长度属性、可选的垂直长度属性、可选的角度属性和可选的水平比率属性组成。

9.2.10.4.5.2 定义属性

<forward>语句的属性定义如下。

- a) horizontal\_length:此属性定义从第一个位置到键合区的水平长度,如图 62 所示。长度的单位由<unit>语句中的<distance>语句定义。
- b) vertical\_length:此属性定义从第一个位置到键合区的垂直长度,如图 62 所示。长度的单位由<unit>语句中的<distance>语句定义。
- c) angle:此属性定义从第一个位置到第二个位置的角度。角度的单位由<unit>语句中的<angle>语句定义。
- d) horizontal\_ratio:此属性定义图 62 中水平长度(L1)和总长度(L)的比率。

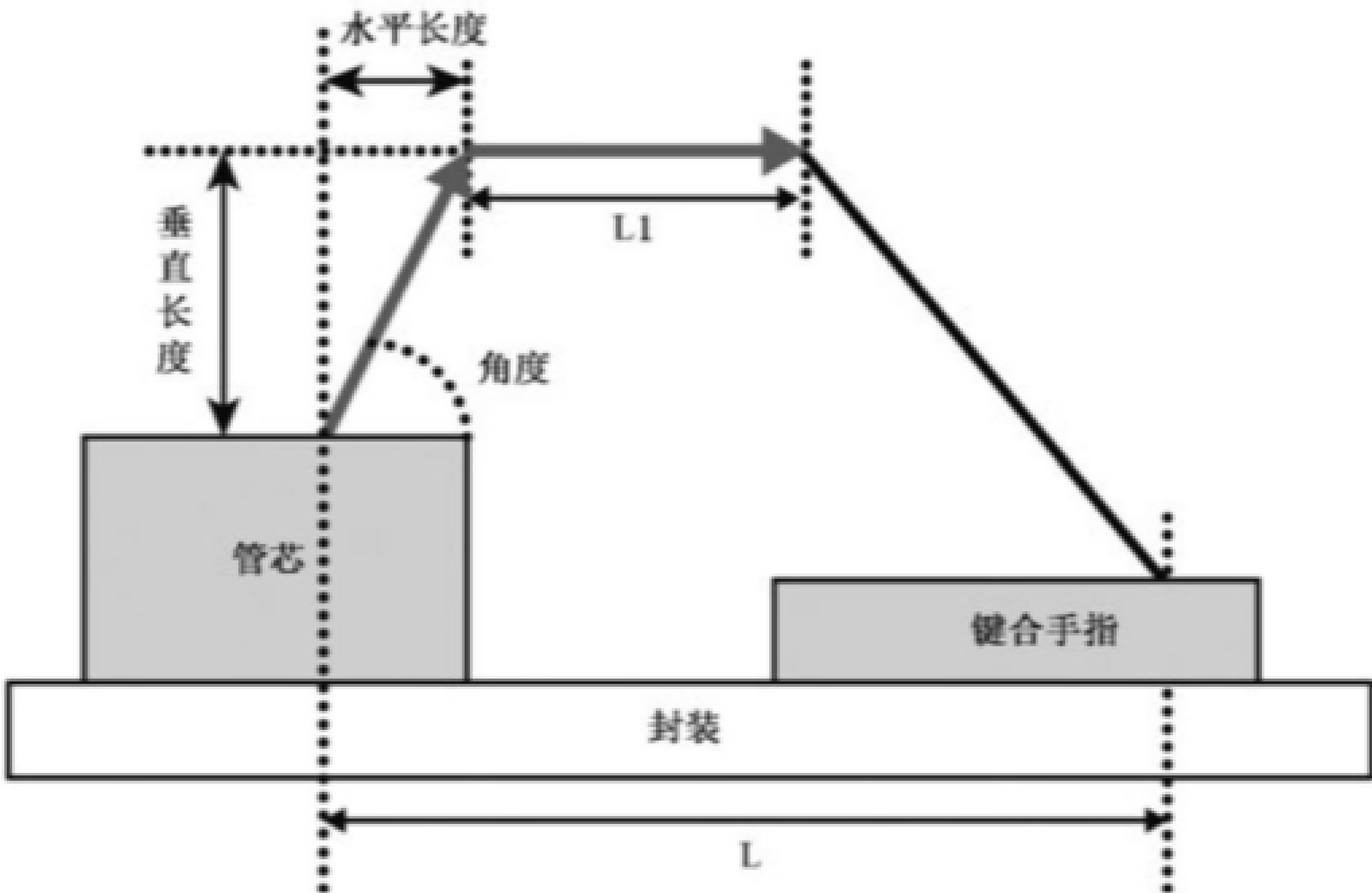


图 62 键合丝形状示意图

9.2.10.4.5.3 例句

以下是<forward>语句的一个示例。

```
<forward horizontal_length="0" vertical_length="100"/>
<forward vertical_length="0" horizontal_ratio="0.125"/>
```

9.2.10.4.6 Backward 语句

9.2.10.4.6.1 概述

<backward>语句定义键合区的键合丝的循环。

```
<backward
    horizontal_length="horizontal_length "
    vertical_length="vertical_length "
    angle="die_side_angle "
```

```
horizontal_ratio="horizontal_ratio "  
/>
```

从第一个位置到管芯的第二个位置指定了一些 <backward> 语句(参见图 63)。<backward> 语句由可选的水平长度属性、可选的垂直长度属性、可选的角度属性和可选的水平比率属性组成。

9.2.10.4.6.2 定义属性

<backward> 语句的属性定义如下。

- a) horizontal\_length: 此属性定义到管芯位置的水平长度, 如图 63 所示。长度的单位由 <unit> 语句中的 <distance> 语句定义。
- b) vertical\_length: 此属性定义从第一个位置到管芯的垂直长度, 如图 63 所示。长度的单位由 <unit> 语句中的 <distance> 语句定义。
- c) angle: 此属性定义从第一个位置到管芯的角度, 如图 63 所示。角度的单位由 <unit> 语句中的 <angle> 语句定义。
- d) horizontal\_ratio: 此属性定义图 63 中水平长度(L1)和总长度(L)的比率。

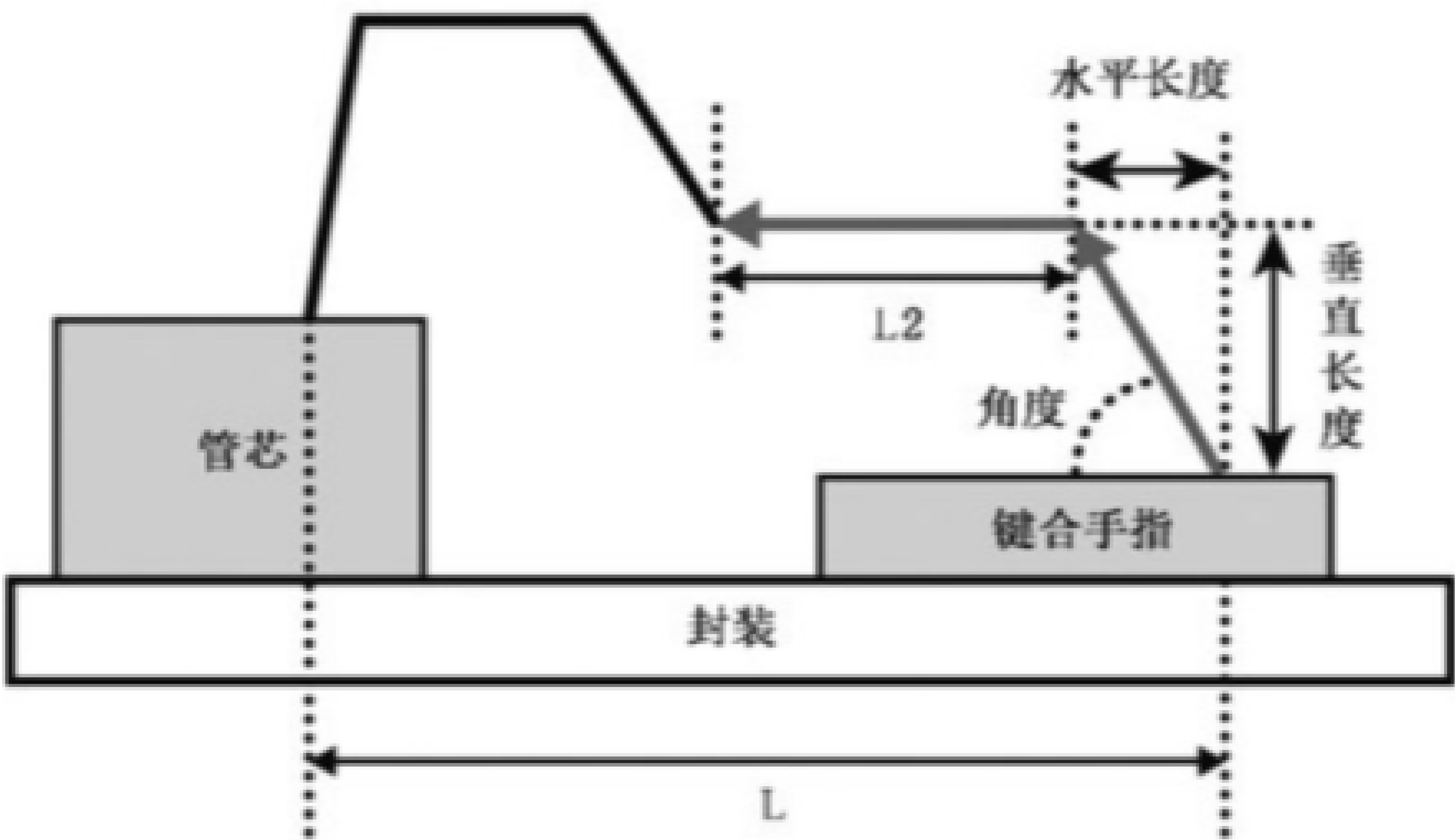


图 63 键合丝形状示意图

9.2.10.4.6.3 例句

以下是 <backward> 语句的示例。

```
<backward angle="20" horizontal_ratio="0.5"/>
```

9.2.10.4.7 length 语句

9.2.10.4.7.1 概述

<length> 语句定义连接线的最小长度和最大长度。

```
<length  
min="minimum_length "  
max="maximum_length "  
/>
```

<length> 语句由 min 属性和 max 属性组成。



9.2.10.4.7.2 定义属性

- ⟨length⟩语句的属性定义如下：
- a) min:此属性定义键合丝的最小长度；
  - b) max:此属性定义键合丝的最大长度。

长度的单位由⟨unit⟩语句中的⟨distance⟩语句定义。

9.2.10.4.7.3 例句

以下是⟨length⟩语句的示例。

```
⟨length min="500" max="3000"/⟩
```

9.2.11 ball\_def 语句

9.2.11.1 概述

⟨ball\_def⟩语句指定焊球的材料和形状。

```
⟨ball_def⟩
    {⟨ball⟩ element}...
    [⟨extensions⟩element]...
⟨/ball_def⟩
```

⟨ball\_def⟩语句由一个或多个⟨ball⟩语句和零个或多个⟨extensions⟩语句组成。

9.2.11.2 语句构成

⟨ball\_def⟩语句包含以下内容：

```
⟨ball⟩
⟨extensions⟩
```

9.2.11.3 例句

以下是⟨ball\_def⟩语句的一个示例。

```
⟨ball_def⟩
    ⟨ball name="BGA_Ball" material="SOLDER"⟩
        ⟨frustum height="250" diam1="300" diam2="300"/⟩
    ⟨/ball⟩
⟨/ball_def⟩
```

9.2.11.4 ball 语句

9.2.11.4.1 概述

⟨ball⟩语句指定焊球的材料和形状。

```
<ball
    name="ball_name "
    [material="material_name "]
>
    {<frustum> element}>...
</ball>
```

<ball>语句由名字属性、可选的材料属性和一个或多<frustum>语句组成。

9.2.11.4.2 定义属性

- <ball>语句的属性定义如下：
- a) name:此属性定义焊球的名称；
  - b) material:此属性定义焊球的材料。

9.2.11.4.3 语句构成

<ball>语句包含以下内容：

```
<frustum>
```

9.2.11.4.4 例句

以下是<ball>语句的一个示例。

```
<ball name="BGA_Ball" material="SOLDER">
    <frustum height="250" diam1="300" diam2="300"/>
</ball>
```

9.2.11.4.5 frustum 语句

9.2.11.4.5.1 概述

<frustum>语句定义焊球的几何形状。

```
<frustum
    height="height "
    diam1="diameter_of_top_side "
    diam2="diameter_of_bottom_side "
/>
```

焊球的形状由一组或多个圆柱体组成，如图 64 所示，柱体从上到下堆放。<frustum>语句由高度属性、dam 1 属性和 dam 2 属性组成。

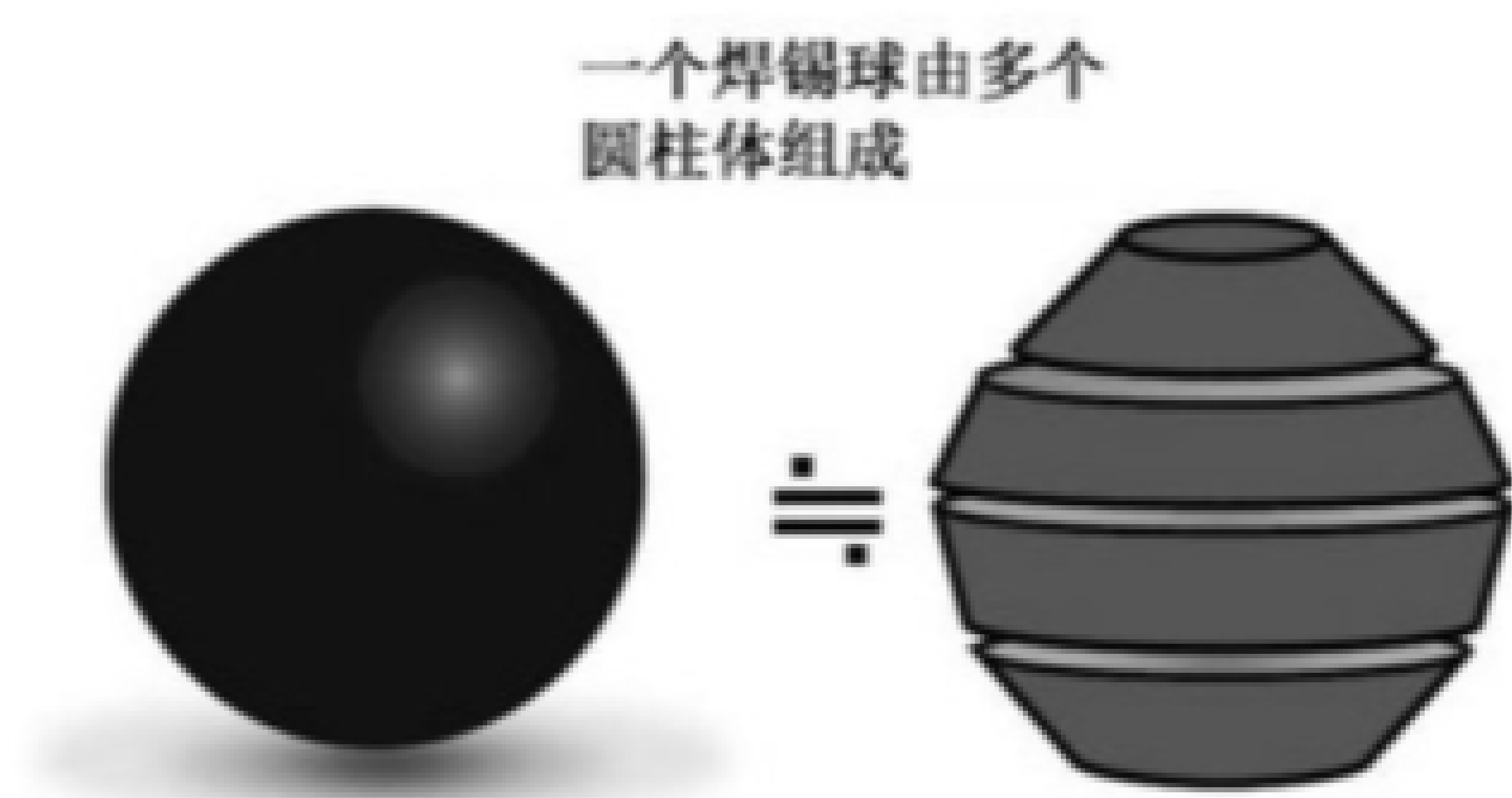


图 64 球表示法示例

9.2.11.4.5.2 定义属性

〈frustum〉语句的属性定义如下：

- a) height：此属性定义柱体的高度，高度的单位由〈unit〉语句中的〈distance〉语句定义；
- b) diam1：此属性定义柱体顶部的直径，直径的单位由〈unit〉语句中的〈distance〉语句定义；
- c) diam2：此属性定义柱体底部的直径，直径的单位由〈unit〉语句中的〈distance〉语句定义。

图 65 显示了〈frustum〉的维度属性之间的关系：高度、直径 1 和直径 2。

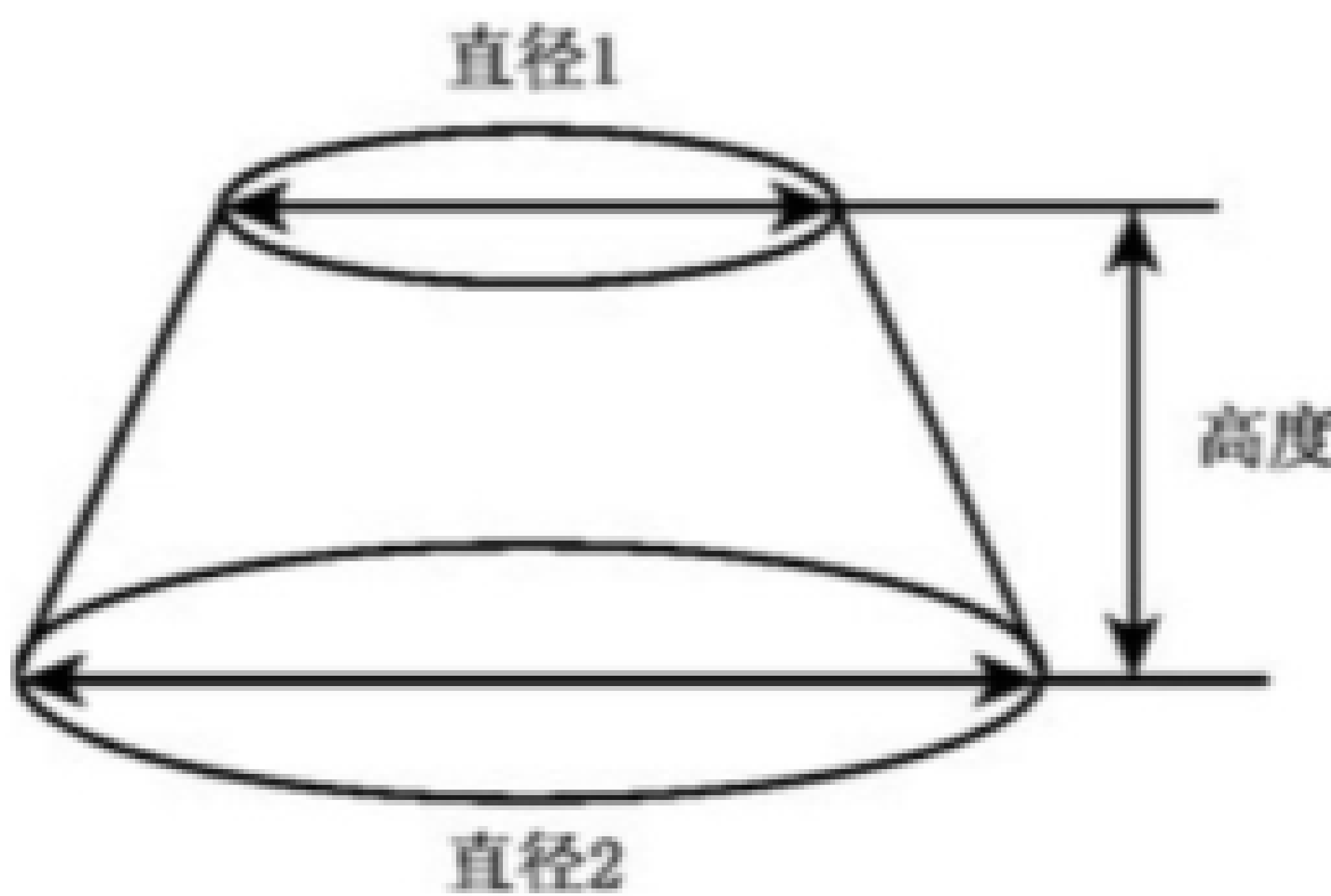


图 65 圆柱体形状示意图

9.2.11.4.5.3 例句

以下是〈frustum〉语句的一个示例。

```
<ball name="BGA_Ball" material="SOLDER">  
  <frustum height="250" diam1="300" diam2="300"/>  
</ball>
```

9.2.12 mold 语句

9.2.12.1 概述

〈mold〉语句定义封装塑模的几何形状和材料。

```
<mold  
  width="width "  
  depth="depth "  
  height="height "  
  material="material_name "  
>  
  [〈extensions〉element]...
```

</mold>

<mold>语句由宽度属性、深度属性、高度属性和材料属性组成。图 66 解释了与叠层相关的<mold>语句的高度属性。

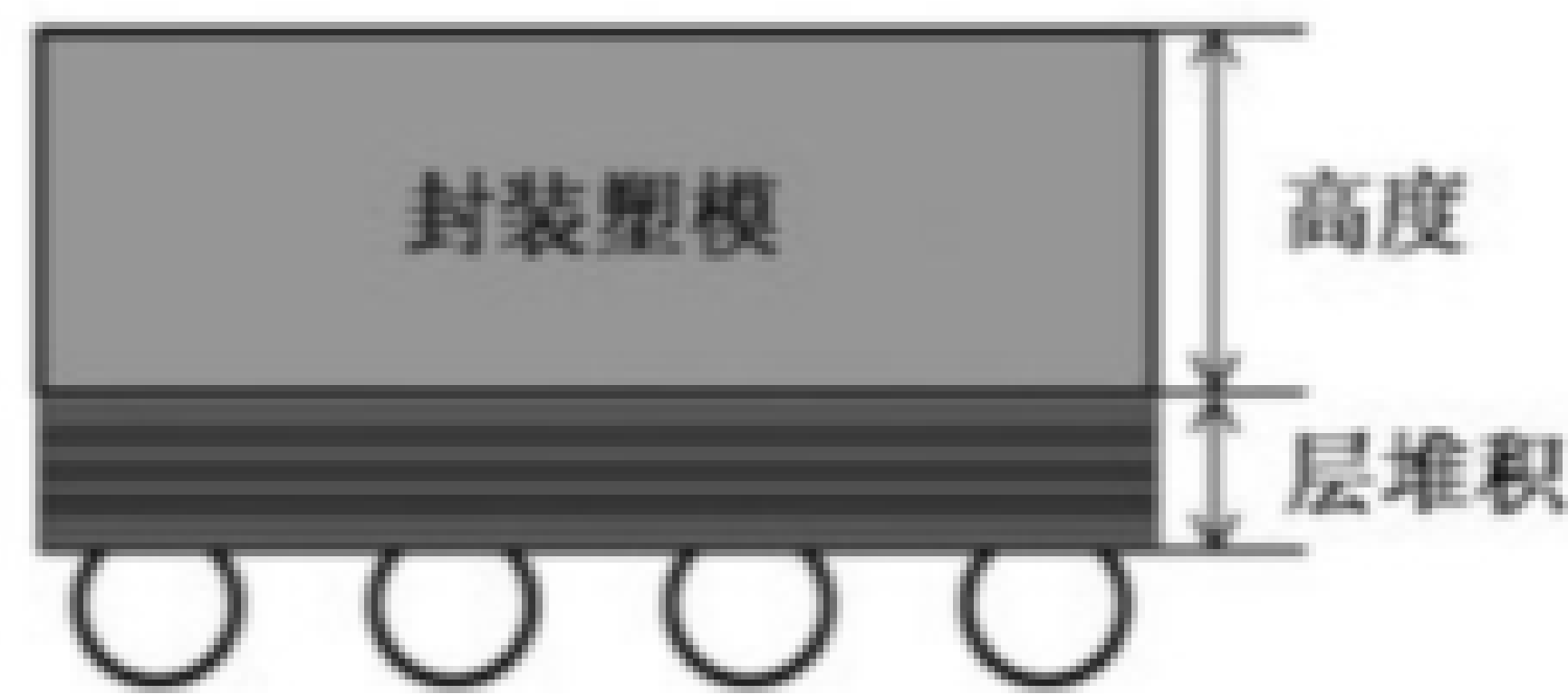


图 66 封装塑模高度属性示意图

9.2.12.2 定义属性

<mold>语句的属性定义如下：

- a) width:此属性定义封装塑模的宽度,如图 67 所示；
- b) depth:此属性定义封装塑模的深度,如图 67 所示；
- c) height:此属性定义封装塑模的高度,如图 67 所示；
- d) material:此属性定义封装塑模的材料。

图 67 显示了<mold>语句的宽度、深度和高度属性。

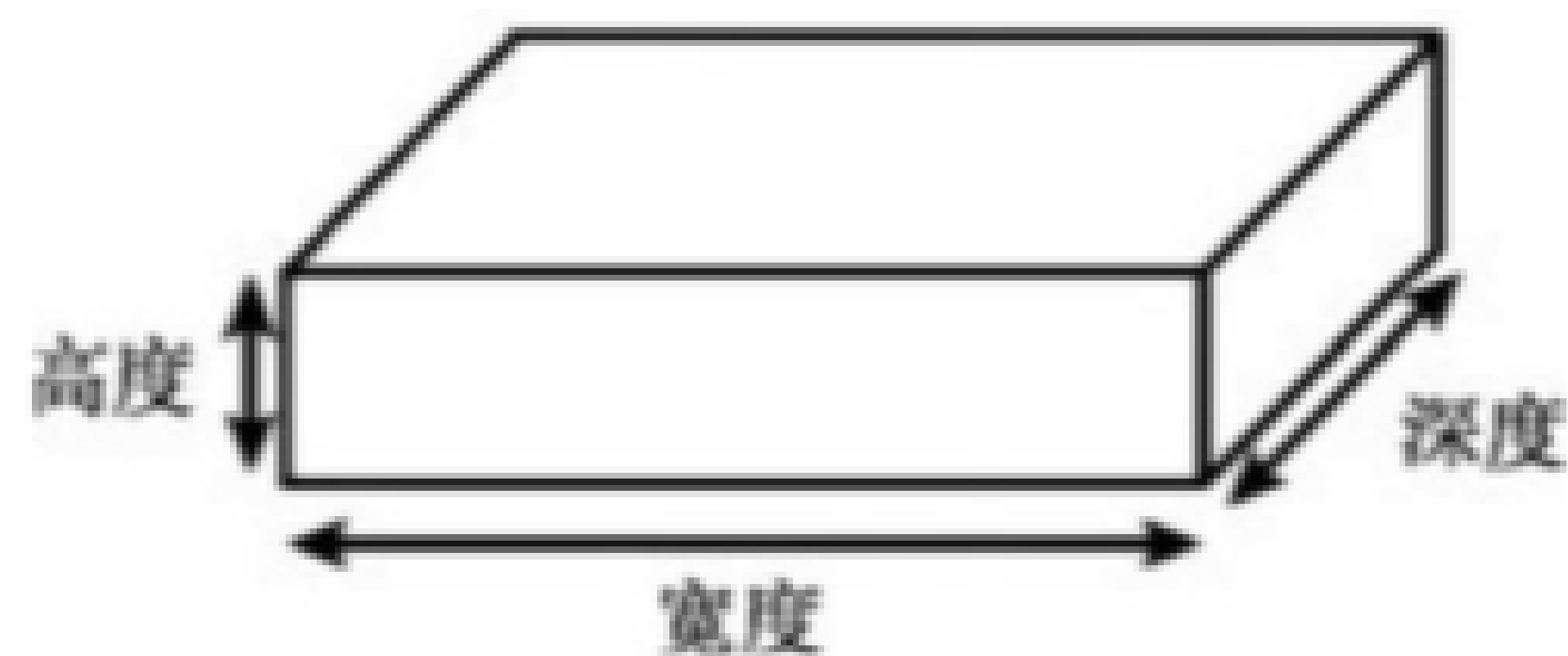


图 67 封装塑模形状示意图

9.2.12.3 语句内容

<mold>语句包含以下内容：

<extensions>

9.2.12.4 例句

以下是<mold>语句的一个示例。

```
<mold
  width="12000"
  depth="12000"
  height="600"
  material="RESIN"
/>
```

9.2.13 conductor\_struct 语句

9.2.13.1 概述

<conductor\_struct>语句定义导体的横截面。

```
<conductor_struct>
    [<trapezoidal_angle> element]...
    [<surface_roughness> element]...
    [<extensions> element]...
</conductor_struct>
```

导体的截面形状定义为具有指定角度的梯形, <conductor\_struct> 语句由零或多个 <trapezoidal\_angle> 语句和零或多个 <surface\_roughness> 语句组成。

9.2.13.2 语句构成

<conductor\_struct> 语句包含以下内容:

```
<trapezoidal_angle>
<surface_roughness>
<extensions>
```

9.2.13.3 例句

以下是 <conductor\_struct> 语句的一个示例。

```
<conductor_struct>
    <trapezoidal_angle layer="L1" angle="60"/>
    <trapezoidal_angle layer="L4" angle="-60"/>
    <surface_roughness layer="L1" UP_RMS="2" DOWN_RMS="5"/>
    <surface_roughness layer="L4" UP_RMS="5" DOWN_RMS="2"/>
</conductor_struct>
```

9.2.13.4 trapezoidal\_angle 语句

9.2.13.4.1 概述

<trapezoidal\_angle> 语句指定导体的截面形状。

```
<trapezoidal_angle
    layer="layer_name "
    angle="angle "
/>
```

<trapezoidal\_angle> 语句由层属性和角属性组成。

9.2.13.4.2 定义属性

<trapezoidal\_angle 语句的属性定义如下:

- a) layer: 此属性定义在 <layer\_def> 语句中指定的导体层的名称。
- b) angle: 导线的截面形状定义为梯形, 此属性指定梯形的角度。角度的单位由 <unit> 语句中的 <angle> 语句定义。

图 68 显示了导体的各种梯形截面形状。

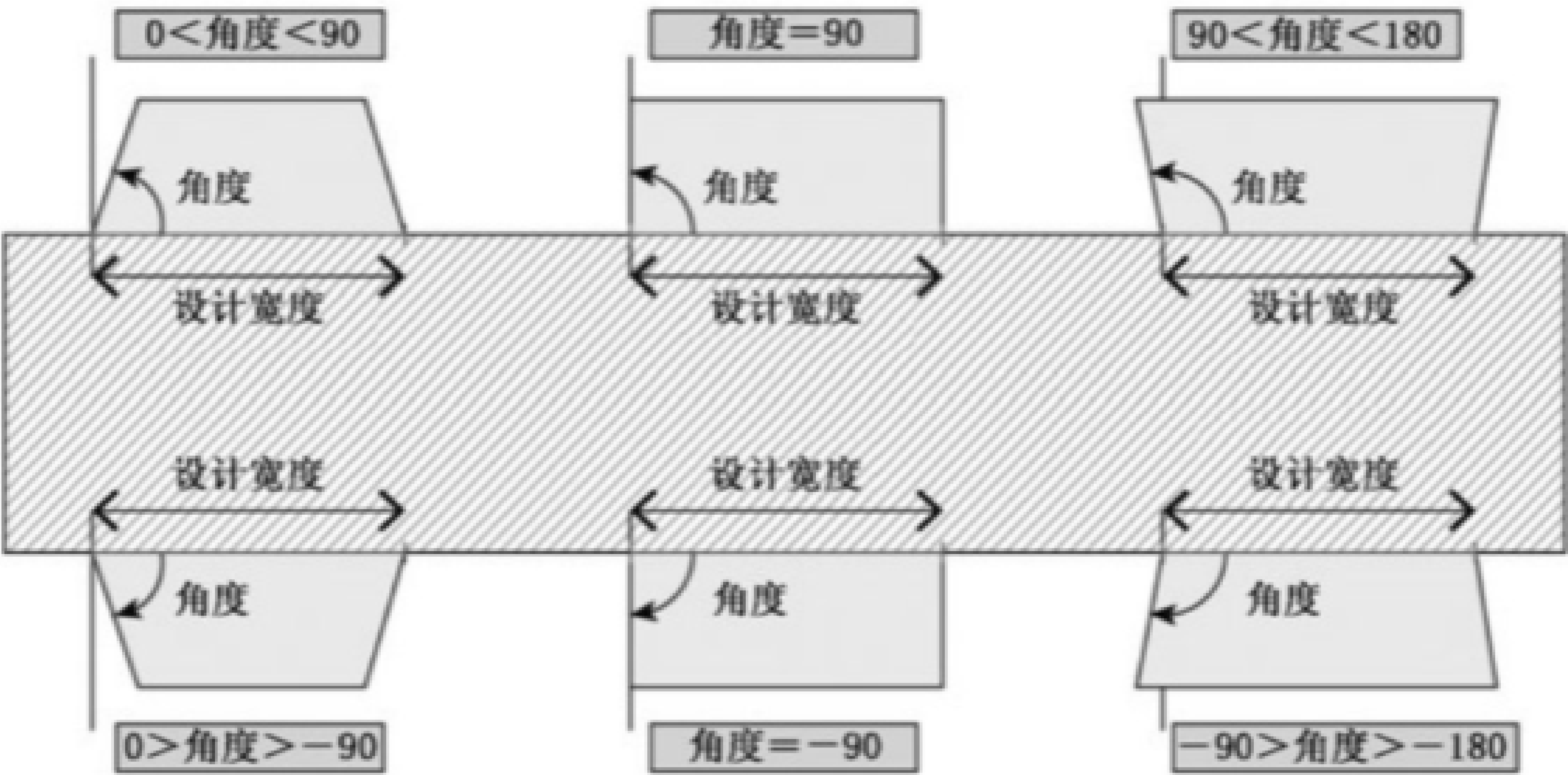


图 68 梯形角示意图

9.2.13.4.3 例句

以下是<trapezoidal\_angle>语句的一个示例。

```
<trapezoidal_angle layer="L1" angle="60"/>
<trapezoidal_angle layer="L4" angle="-60"/>
```

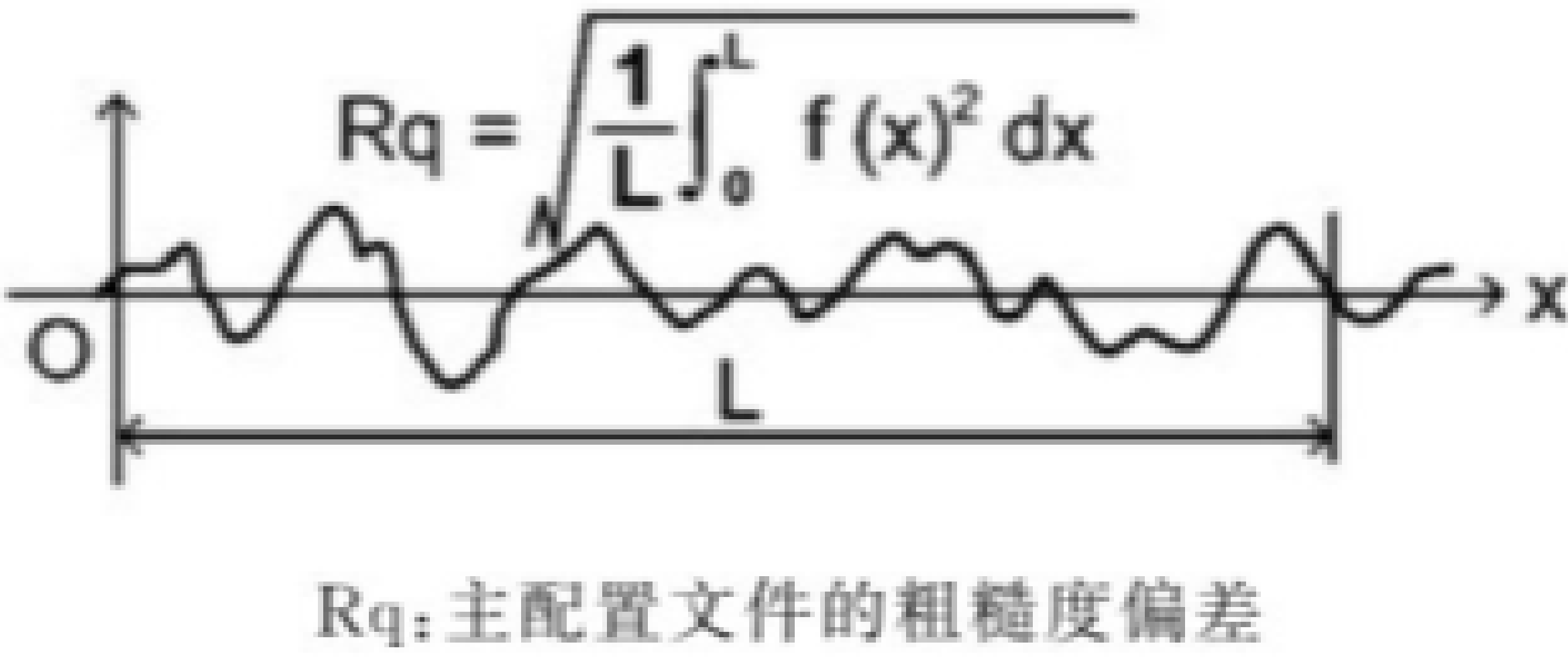
9.2.13.5 surface\_roughness 语句

9.2.13.5.1 概述

<surface\_roughness>语句定义导体表面的粗糙度。

```
<surface_roughness
  layer="layer_name "
  UP_RMS="up_rms "
  DOWN_RMS="down_rms "
/>
```

<surface\_roughness>语句由 layer\_name 属性、UP\_RMS 属性和 DOWN\_RMS 属性组成。图 69 以均方根表示导体表面的粗糙度。



Rq:主配置文件的粗糙度偏差

图 69 表面粗糙度示意图

9.2.13.5.2 定义属性

<surface\_roughness>语句的属性定义如下：

- a) layer:此属性定义在<layer\_def>语句中指定的导体层的名称；
- b) UP\_RMS:此属性定义 RMS 中导体顶部表面的粗糙度；

c) DOWN\_RMS:此属性定义 RMS 中导体底部表面的粗糙度。

图 70 显示了导体层、导体顶部表面的粗糙度和导体底部表面的粗糙度,它们都是<surface\_roughness>语句的属性。

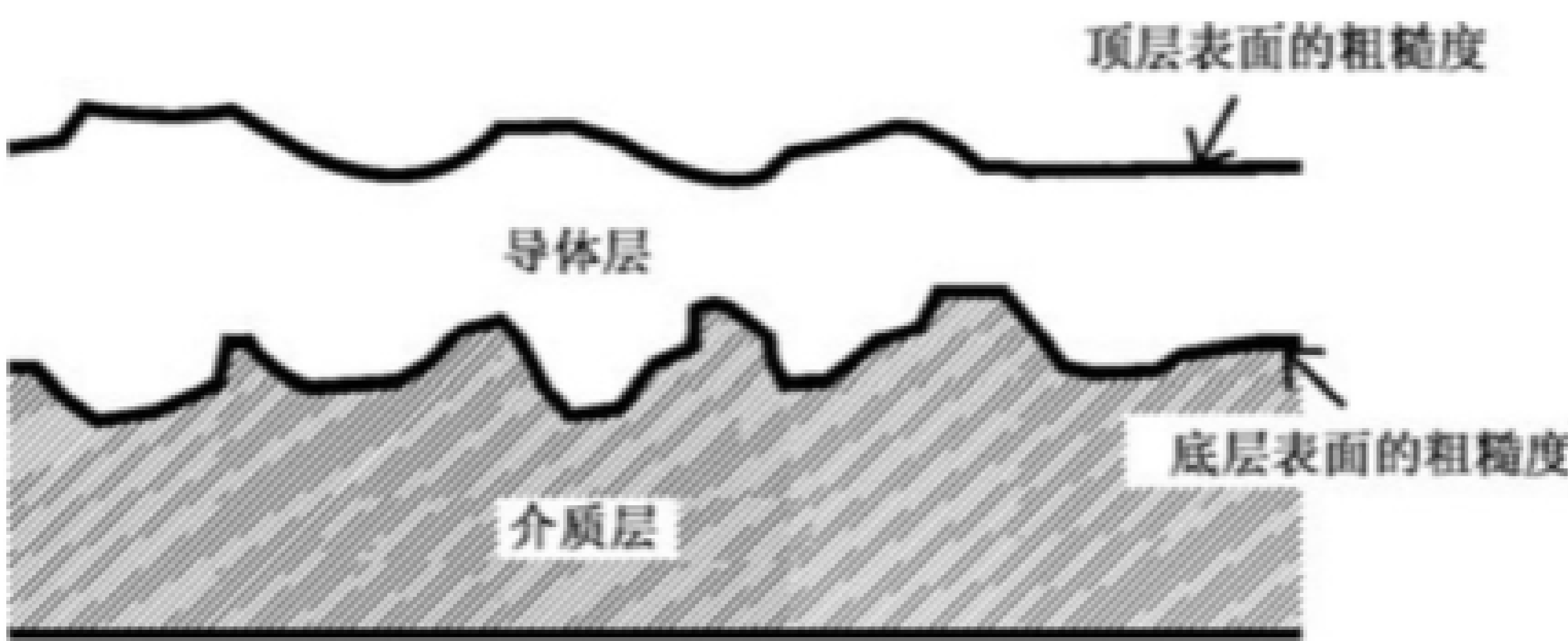


图 70 UP\_RMS 和 DUP\_RMS 示意图

9.2.13.5.3 例句

以下是<surface\_roughness>语句的一个示例。

```
<surface_roughness layer="L1" UP_RMS="2" DOWN_RMS="5"/>
<surface_roughness layer="L4" UP_RMS="5" DOWN_RMS="2"/>
```

9.3 Constrainerule 语句

9.3.1 概述

<Constrainerule>语句,通过<height\_limit>语句定义元器件的限制高度,通过<blocking>语句指定布局路由禁止区域,并通过<design\_rule\_area>语句定义非默认的设计规则区域。

```
<Constrainerule>
  [<height_limit> element]...
  [<blockage> element]...
  [<design_rule_area> element]...
  [<extensions> element]...

</Constrainerule>
```

<Constrainerule>语句由零或多个<height\_limit>语句、零或多个<design\_rule\_area>语句和零个或多个<design\_rule\_area>语句组成。

9.3.2 语句构成

<Constrainerule>语句包含以下内容:

```
<height_limit>
<blockage>
<design_rule_area>
<extensions>
```

## 9.3.3 例句

以下是〈Constrainrule〉语句的一个示例。

```

〈Constrainrule〉
  〈height_limit〉
    〈top
      name="Top_Connector_Location"
      height="200"
      shape_id="Rec_6000 μm_8000 μm"
      x="3000" y="2000"
    /〉
    〈bottom
      name="Bottom_Connector_Location"
      height="200"
      shape_id="Rec_6000 μm_8000 μm"
      x="3000" y="2000"
    /〉
  /〉height_limit〉
  〈blockage〉
    〈placement
      layer="11"
      shape_id="kout_top_200 μm_200 μm "
      x="100"
      y="100"
    /〉
    〈routing
      layer="11"
      shape_id="kout_bottom_200 μm_200 μm "
      x="100"
      y="100"
    /〉
    〈routing
      layer="14"
      shape_id="kout_bottom_400 μm_400 μm "
      x="0"
      y="0"
    /〉
  /〉blockage〉
  〈design_rule_area
    ref_rule_name="C4_Area"
    shape_id="SQ_6000 μm_6000 μm"
    x="0" y="0"
  /〉

```



```
priority="10"  
/  
</Constrainrule>
```

9.3.4 height\_limit 语句

9.3.4.1 概述

<height\_limit>语句定义从顶层或底层的表面到组件的高度限制。

```
<height_limit>  
    [<top> element]...  
    [<bottom> element]...  
    [<extensions> element]...  
</height_limit>
```

<height\_limit>语句由零或多个<top>语句、零或多个<bottom>语句和零或多个<extensions>组成,图 71 显示了来自顶层或底层表面的组件的高度限制。

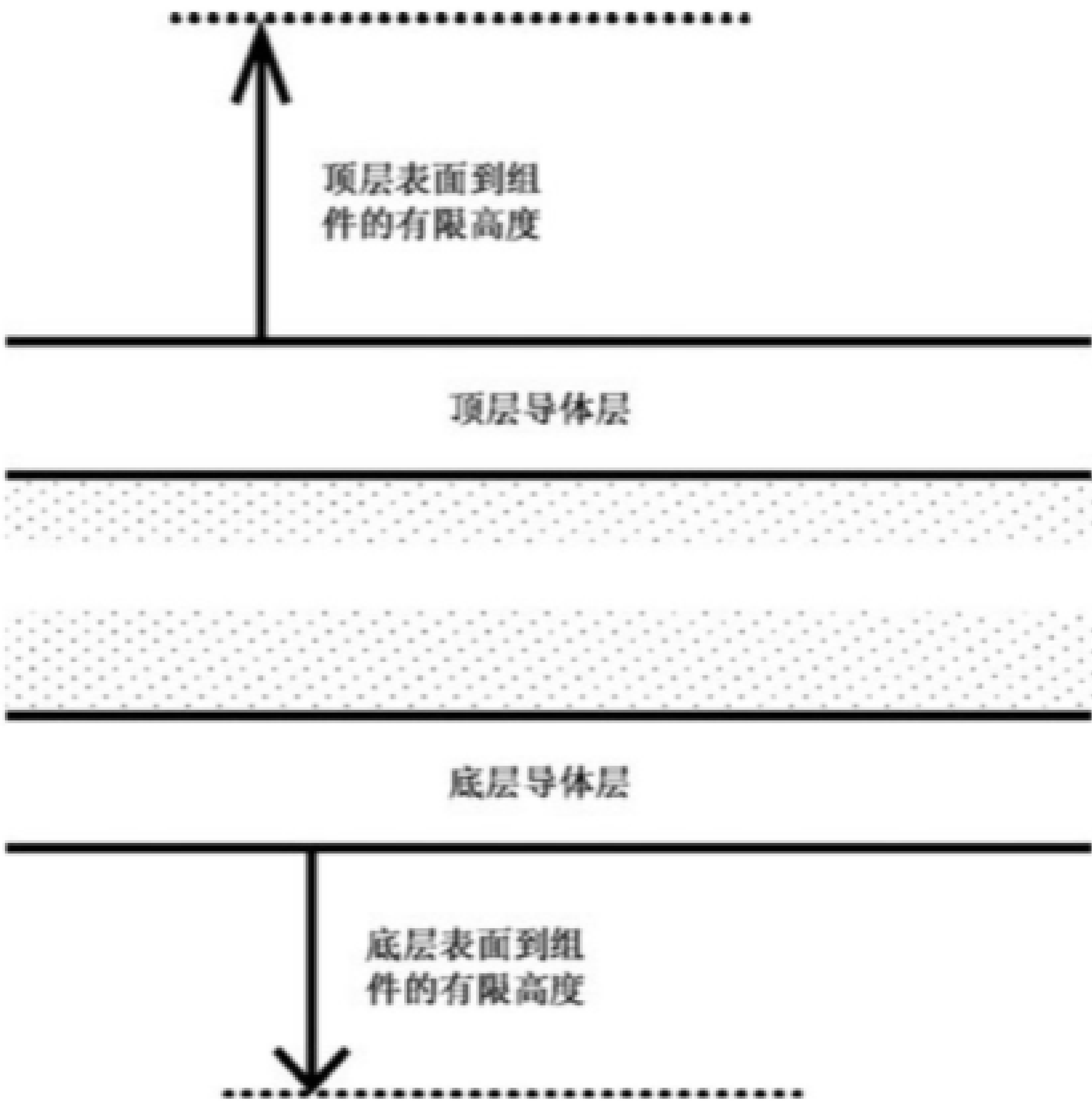


图 71 顶高和底高限制示意图

9.3.4.2 语句构成

<height\_limit>语句包含以下内容:

- <top>
- <bottom>
- <extensions>

9.3.4.3 例句

以下是<height\_limit>语句的一个示例。

```
<height_limit>
  <top
    name="Top_Connector_Location"
    height="200"
    shape_id="Rec_6000 μm_8000 μm"
    x="3000" y="2000"
  />
  <bottom
    name="Bottom_Connector_Location"
    height="200"
    shape_id="Rec_6000 μm_8000 μm"
    x="3000" y="2000"
  />
</height_limit>
```

9.3.4.4 top 语句

9.3.4.4.1 概述

<top>语句定义顶层上放置的组件的最大高度。

```
<top
  name="area_name "
  height="height_limitation "
  shape_id="identifier_of_referenced_shape "
  x="x_coordinate "
  y="y_coordinate "
  [angle="rotation_angle "]
/>
```

<top>语句由 name 属性、height 属性、shape\_id 属性、x 属性、y 属性和可选 angle 属性组成。

9.3.4.4.2 定义属性

<top>语句的属性定义如下。

- a) name:此属性定义顶层高度限制的区域的名称。
- b) height:此属性定义顶层的最大高度。高度的单位由<unit>语句中的<distance>语句定义。
- c) shape\_id:此属性定义在顶层具有指定高度限制空间的形状的标识号,形状的原点坐标与高度限制空间的原点坐标一致。
- d) x:此属性定义高度限制区域中原点的 x 坐标。坐标的单位由<unit>语句中的<distance>语句定义。
- e) y:此属性定义高度限制区域中原点的 y 坐标。坐标的单位由<unit>语句中的<distance>语句定义。
- f) angle:此属性定义逆时针旋转相对于局部原点的角度,如果未指定,则将零设置为默认值。角

度的单位由<unit>语句中的<distance>语句定义。

9.3.4.4.3 例句

以下是<top>语句的示例。

```
<top
  name="Top_Connector_Location"
  height="200"
  shape_id="Rec_6000 μm_8000 μm"
  x="3000" y="2000"
/>
```

9.3.4.5 bottom 语句

9.3.4.5.1 概述

<bottom>语句定义底层放置的组件的最大高度。

```
<bottom
  name="area_name "
  height="height_limitation "
  shape_id="identifier_of_referenced_shape "
  x="x_coordinate "
  y="y_coordinate "
  [angle="rotation_angle "]
/>
```

<bottom>语句由 name 属性、height 属性、shape\_id 属性、x 属性、y 属性和可选 angle 属性组成。

9.3.4.5.2 定义属性

<bottom>语句的属性定义如下。

- a) name:此属性定义底层高度限制的区域名称。
- b) height:此属性定义底层的最大高度。高度的单位由<unit>语句中的<distance>语句定义。
- c) shape\_id:此属性定义在底层具有指定高度限制空间的形状的标识号,形状的原点坐标与高度限制空间的原点坐标一致。
- d) x:此属性定义高度限制区域中原点的 x 坐标。坐标的单位由<unit>语句中的<distance>语句定义。
- e) y:此属性定义高度限制区域中原点的 y 坐标。坐标的单位由<unit>语句中的<distance>语句定义。
- f) angle:此属性定义逆时针旋转相对于局部原点的角度,如果未指定,则将零设置为默认值。角度的单位由<unit>语句中的<distance>语句定义。

9.3.4.5.3 例句

以下是<bottom>语句的一个示例。

```
<bottom
    name="Bottom_Connector_Location"
    height="200"
    shape_id="Rec_6000 μm_8000 μm"
    x="3000" y="2000"
/>
```

9.3.5 blockage 语句

9.3.5.1 概述

<blockage>语句指定组件安装和/或导电物体布局的禁止区域。

```
<blockage>
    [<placement> element]...
    [<routing> element]...
    [<extensions> element]...
</blockage>
```

<blockage>语句由零个或多个<placement>、<routing>和<extensions>语句组成。

9.3.5.2 语句构成

<blockage>语句包含以下内容：

```
<placement>
<routing>
<extensions>
```

9.3.5.3 例句

以下是<blockage>语句的一个示例。

```
<blockage>
    <placement
        ref_layer="L1"
        shape_id="kout_top_200 μm_200 μm"
        x="100"
        y="100"
    />
    <routing
        ref_layer="L1"
        shape_id="kout_top_200 μm_200 μm"
        x="100"
        y="100"
    />
    <routing
```

```
ref_layer="L4"
shape_id="kout_top_400 μm_400 μm"
x="0"
y="0"
/>
</blockage>
```

9.3.5.4 placement 语句

9.3.5.4.1 概述

<placement>语句指定用于禁止组件安装的层和形状。

```
<placement
    ref_layer="name_of_referenced_layer "
    [attach="layer_side "]
    shape_id="identifier_of_referenced_shape "
    x="x_coordinate "
    y="y_coordinate "
    [angle="rotation_angle "]
/>
```

<placement>语句由 ref\_layer 属性、shape\_id 属性、x 属性、y 属性和 angle 属性组成。

9.3.5.4.2 属性定义

<placement>语句的属性定义如下。

- a) ref\_layer:该属性指定在<ref\_layer>语句中指定的导体层的名称。
- b) attach:该属性通过关键字“ABOVE”或“BELOW”指定要构造放置填充区域的层侧。如果指定了“ABOVE”,则在指定层上方构造填充区域。如果指定了“BELOW”,则在指定层下方构造填充区域。
- c) shape\_id:此属性指定在<shape>语句中定义的预定义形状的标识符。参考形状应在同一文件中的<shape>语句处定义。
- d) x:该属性指定原点的 x 坐标。坐标的单位由<unit>语句中的<distance>语句定义。
- e) y:该属性指定原点的 y 坐标。坐标的单位由<unit>语句中的<distance>语句定义。
- f) angle:该属性指定相对于局部原点的逆时针旋转角度。如果未指定,则将零设置为默认值。角度的单位由<unit>语句中的<angle>语句定义。

9.3.5.4.3 例句

以下是<placement>语句的一个示例。

```
<placement
    ref_layer="L1"
    shape_id="keep_out_200 μm_200 μm"
    x="100"
```

```
        y="100"  
    />
```

9.3.5.5 routing 语句

9.3.5.5.1 概述

⟨routing⟩语句指定用于禁止导电对象(如迹线、过孔和平面)布局的层和形状。

```
⟨routing  
    ref_layer="name_of_referenced_layer "  
    shape_id="identifier_of_referenced_shape "  
    x="x_coordinate "  
    y="y_coordinate "  
    [angle="rotation_angle "]  
/>
```

⟨routing⟩语句由 ref\_layer 属性、shape\_id 属性、x 属性、y 属性和 angle 属性组成。

9.3.5.5.2 属性定义

⟨routing⟩语句的属性定义如下。

- a) ref\_layer:该属性指定在⟨layer\_def⟩语句中指定的导体层的名称。
- b) shape\_id:此属性指定在⟨shape⟩语句中定义的预定义形状的标识符。参考形状应在同一文件中的⟨shape⟩语句处定义。
- c) x:该属性指定原点的 x 坐标。坐标的单位由⟨unit⟩语句中的⟨distance⟩语句定义。
- d) y:该属性指定原点的 y 坐标。坐标的单位由⟨unit⟩语句中的⟨distance⟩语句定义。
- e) angle:该属性指定相对于局部原点的逆时针旋转角度。如果未指定,则将零设置为默认值。角度的单位由⟨unit⟩语句中的⟨angle⟩语句定义。

9.3.5.5.3 例句

以下是⟨routing⟩语句的一个示例。

```
⟨routing  
    ref_layer="L1"  
    shape_id="kout_top_200 μm_200 μm"  
    x="100"  
    y="100"  
/>
```

9.3.6 design\_rule\_area 语句

9.3.6.1 概述

⟨design\_rule\_area⟩定义具有非默认设计规则的特定区域。

```
⟨design_rule_area  
    ref_rule_name="name_of_referenced_design_rule "
```

```

    shape_id="identifier_of_referenced_shape "
    x="x_coordinate "
    y="y_coordinate "
    [angle="rotation_angle "]
    [priority=" priority "]
>

    [<extensions> element]...
</design_rule_area>
```

此区域中的默认设计规则被非默认设计规则覆盖,可在<design\_rule\_area>中定义与默认设计规则不同的规则,但不可能在本地定义叠层,即使本地设计规则包含定义层结构的语句,定义也将被忽略。<design\_rule\_area>语句由 ref\_rule\_name 属性、shape\_id 属性、x 属性、y 属性和可选 angle 属性以及可选的优先级属性组成。图 72 显示了默认设计规则区域和非默认设计规则区域之间的关系。

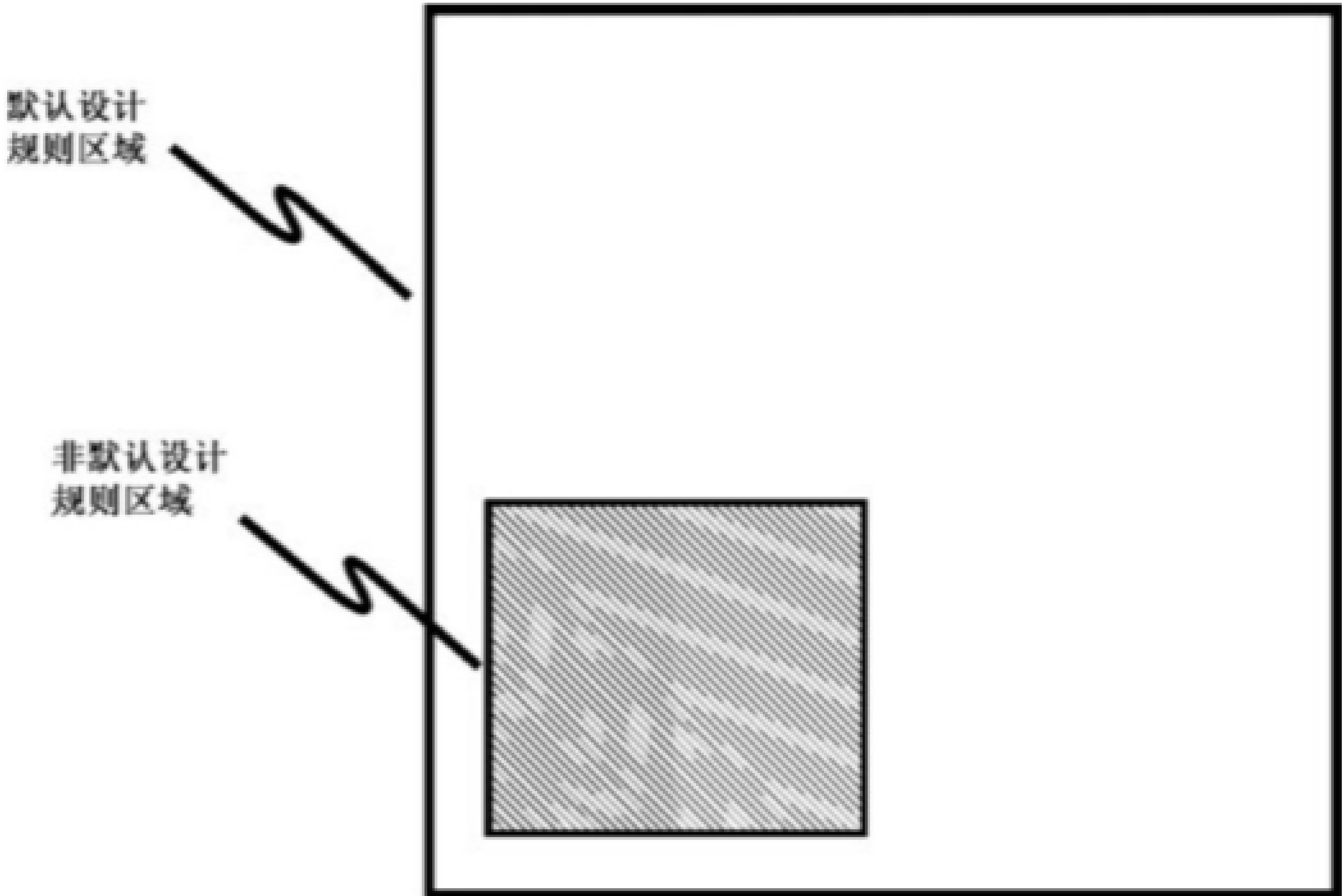


图 72 默认设计规则区域与非默认设计规则区域关系示意图

9.3.6.2 定义属性

<design\_rule\_area>语句的属性定义如下。

- a) ref\_rule\_name:此属性定义在非默认设计规则区域中使用的预定义物理设计规则的名称,物理设计规则在<Physicaldesign>语句中定义。
- b) shape\_id:此属性定义在非默认设计规则区域中指定的形状的标识号,形状的原点坐标与区域的原点坐标一致。
- c) x:此属性定义原点的 x 坐标。坐标的单位由<unit>语句中的<distance>语句定义。
- d) y:此属性定义原点的 y 坐标。坐标的单位由<unit>语句中的<distance>语句定义。
- e) angle:此属性定义逆时针旋转相对于局部原点的角度,如果未指定,则将零设置为默认值。角度的单位由<unit>语句中的<angle>语句定义。
- f) priority:此属性指定由<deisgn\_rule\_area>元素指定的物理设计规则的优先级。数字越大表示优先级越高,默认物理设计区域的优先级为零。如果未指定,则将一个设置为默认值。

9.3.6.3 语句构成

〈design\_rule\_area〉语句包含以下内容：

〈extensions〉

9.3.6.4 例句

以下是〈design\_rule\_area〉语句的一个示例。

```
〈design_rule_area
  ref_rule_name="C4_Area"
  shape_id="SQ_6000 μm_6000 μm"
  x="0" y="0"
  priority="10"
/〉
```

10 N 格式

10.1 N 格式文件目的

N 格式文件不仅包括信号连接,还包括电源/接地网络,N 格式文件符合 Verilog-GHDL。

10.2 如何定义网表中的电源层/接地层

添加 /\* PG\_NET \* /\ /\* PGNET=group\_name \* /\ /\* PWR=group\_name \* /\ /\* GND=group\_name \* /关键字定义端口：

- a) 电源/接地端口方向为外部；
- b) 电源/接地的网络属性是线路,而不是提供 0 或 1；
- c) 在电源/接地端口处添加 PG\_NET 或 PG\_NET=group\_name 关键字作为注释,其中 group\_name 是电源/接地端口的名称；
- d) 在信号端口添加 PWR=roup\_name 和 GND=roup\_name 关键字作为要指定的注释,以指定相关的电源域,其中 group\_name 是电源/接地端口的名称；
- e) group\_name 是区分大小写的。

10.3 示例

以下是 N 格式的示例。

```
module XXXX ( A,B,C,D,AVDD1 , AVDD2,AVSS,DVDD,DVSS)
  input A ; /* PWR=analog_vdd_1.8 GND=analog_gnd */
  output ; /* PWR=analog_vdd_1.8 GND=analog_gnd */
  input C ; /* PWR=digital_vdd_1.2 GND=digital_gnd */
  output D ; /* PWR=digital_vdd_1.2 GND=digital_gnd */
  inout AVDD1; /* PG_NET=analog_vdd_1.8 */
  inout AVDD2; /* PG_NET=analog_vdd_1.8 */
  inout AVSS; /* PG_NET=analog_gnd */
```



```
inout DVDD;/ * PG_NET= digital_1.2 * /
inout DVSS;/ * PG_NET= digital_vss * /
```

11 G 格式

11.1 G 格式 Basics 语言

11.1.1 排版和句法

本节描述了 LPB G 格式的语法定义中使用的约定：

- a) `text`:单空格和粗体字用于定义字面键入的属性或语句；
- b) *italic*:斜体字用于定义用户定义的信息,该信息将被替换为名称或值；
- c) `|`:竖线分隔单个属性或语句的可能选择,它们优先于任何其他字符；
- d) `[ ]`:括号表示可选属性或语句,当与垂直条一起使用时,它们包含一个选项列表；
- e) `...`:三个点表示可以重复先前的值。

注 1: G 格式的所有字符串都是区分大小写的；  
注 2: 本文件中的所有代码示例都是用单空格字体编写的。

11.1.2 预留字符

本节描述了预留字符：

- a) `#`:如果一行的第一个字符是井( # )号,则忽略该行,用于注释；
- b) `“ ”`:两个双引号之间的任何数据都被视为字符串。

11.2 结构

G 格式文件由几个部分组成,每个部分都以一个点(.)开始,后面跟着一个关键字。`.version`、`.unit` 和 `.Scale` 部分各由一行组成,所有其他部分都以 `.end` 结尾,后面跟着相同的关键字,可以省略 `.end` 行中的关键字。标题部分由 `.version`、`.unit` 和 `.Scale` 部分组成,除了 `.version` 部分,任何部分都可以省略。G 格式文件的部分如下：

```
.version x y
.unit [inch | mm]
.scale value

.material
definition
.end [material]

.layer
definition
.end [layer]

.material
definition
.end [material]
```

.layer  
*definition*  
.end [layer]

.shape  
*definition*  
.end [shape]

.board\_geom  
*definition*  
.end [board\_geom]

.padstack  
*definition*  
.end [padstack]

.part  
*definition*  
.end [part]

.component  
*definition*  
.end [component]

.netattr  
*definition*  
.end [netattr]

.netlist  
*definition*  
.end [netlist]

.via  
*definition*  
.end [via]

.bondwire  
*definition*  
.end [bondwire]

.route  
*definition*  
.end [route]

11.3 Header 部分

11.3.1 概述

header 部分由.version、.unit 和.scale 组成。

- a) .version x y:本节指定 G 格式文件的版本,x 是主要版本号,y 是次要版本号,z 表示本文件的版本,z 对应于 C/R/M 格式的版本号。版本部分应出现在文件中的任何其他部分之前。

x        整数  
y        整数  
z        2020

- b) .unit value:本节指定在 G 格式文件中使用的几何单位,数值须为英寸或毫米,如果省略此节或值,则假定英寸为默认值。

value    inch or mm

- c) .scale value:本节指定在 G 格式文件中使用的几何比例尺,文件中数据的实际维数是通过将数字除以刻度值来确定的。例如,如果该单位被设置为英寸和比例尺为 1 000,几何数据是毫升或 1/1 000 英寸,如果单位设置为毫米,规模为 1 000,则几何数据以微米或微米为单位,默认为 1。

value    整数

11.3.2 例句

下面是 G 格式文件的标题部分的示例。

```
.version 1 1    2020  
.unit mm  
.scale 1000
```

11.4 Material 部分

11.4.1 概述

.material 部分由以下内容组成:

```
.material  
definition  
.end [material]
```

这是一个提供材料属性的可选部分,列出设计中使用的材料及其性能。每种材料的定义如下:

C                    materialName conductivity  
or  
D                    materialName permittivity permeability lossTangent

其中,C 代表导电材料,D 代表介电材料。

materialName :材料的名称附上双引号

Conductivity: 电导率(1/Ω · mm)

Permittivity: 相对介电常数或介电常数

Permeability: 相对磁导率

*lossTangent* :介质损耗角正切

11.4.2 例句

下面是.material 部分的一个示例。

```
.material
D "AIR" 1.0 1.0 0.0
C "COPPER" 59000
C "GOLD"45500
D "FR-4" 4.5 1.0 0.035
D "SR" 4.3 1.0 0.03
.end material
```

11.5 Layer 部分

.layer 部分由以下内容组成：

```
.layer
definition
.end [layer]
```

本条描述了从上到下(或从前到后)的叠层。每一层的定义如下：

```
name thickness type conducting dielectric1[ dielectric2 dielectric3]
```

其中：

*name* :是用双引号括起来的图层的名称。如果未知,写“”。

*thickness* :是层的厚度,如果未知,写 0。

*type* :由单个字符定义的层的类型：

- S 信号层
- D 介电层
- P 电源或接地层,它可以区别于信号层

*conducting* :用双引号括起来的字符串的导电材料的名称。

*dielectric1* :用双引号括起来的字符串的介质材料的名称。

注：在整个 G 格式文件中,信号层编号是通过只计算信号/电源/地面层从顶部依次编号的数字。

11.6 Shape 部分

11.6.1 概述

.shape 部分由以下内容组成：

```
.shape
definition
.end [shape]
```

本节定义从同一文件中的其他部分引用的形状,可用的形状有多边形、矩形、方形、圆形、环形、长方

形、指形、弹形和复合形状。

形状定义: id keyword parameters: 其中, id 是将被其他人引用的数字, 从 1 依次编号, 关键字和参数描述如下。

- a) id polygon { x1 y1 x2 y2 ... }: 多边形形状的顶点(见图 73)用{}括起来, 数据可以显示在多行上, 最后一点不必与第一点相同, 参考点在(0,0)。

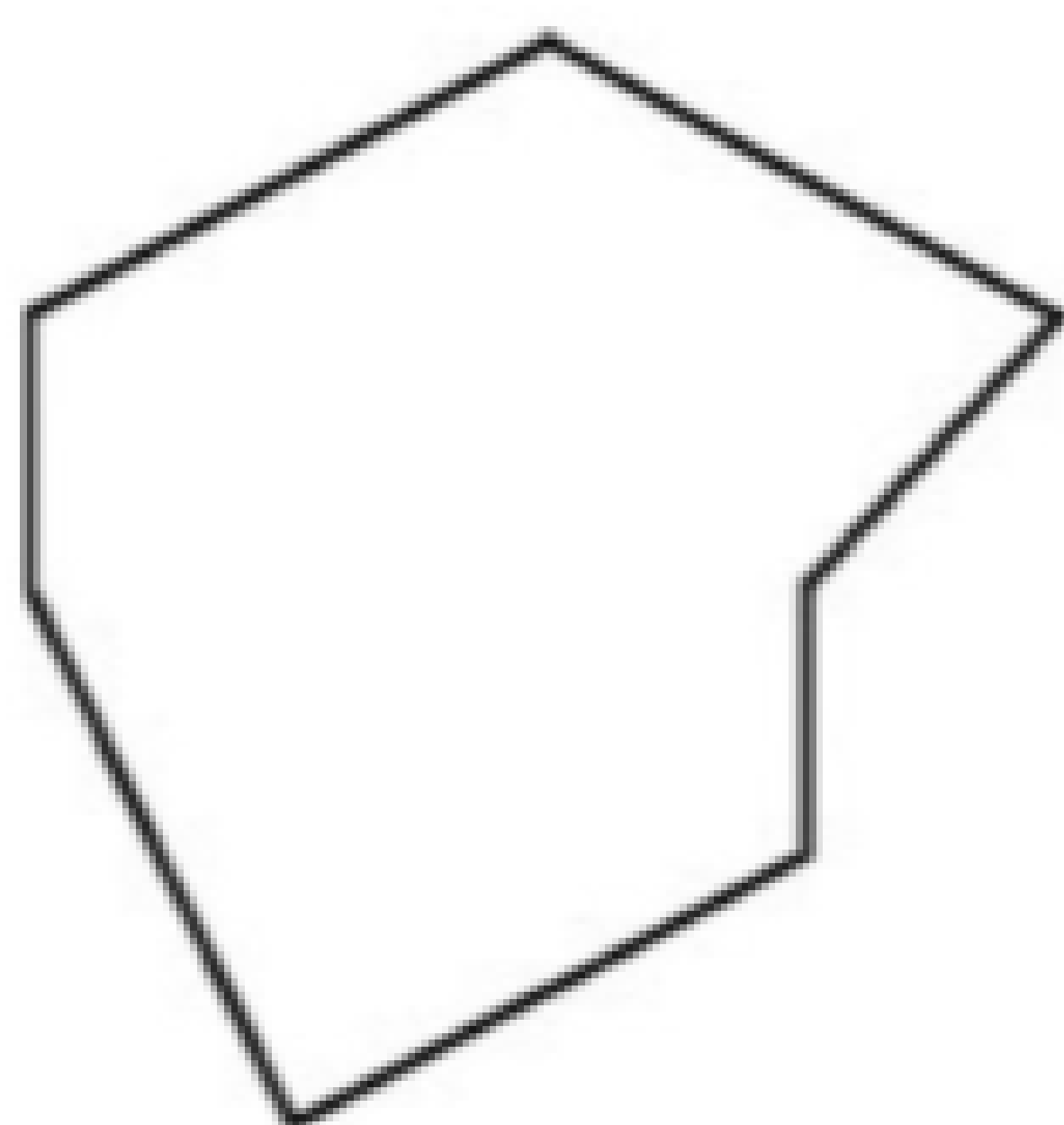


图 73 多边形

- b) id rectangle width height: 每个矩形由宽度和长度定义, 定义应出现在一行上, 参考点位于矩形的中心, 图 74 将参考点标记为 X。

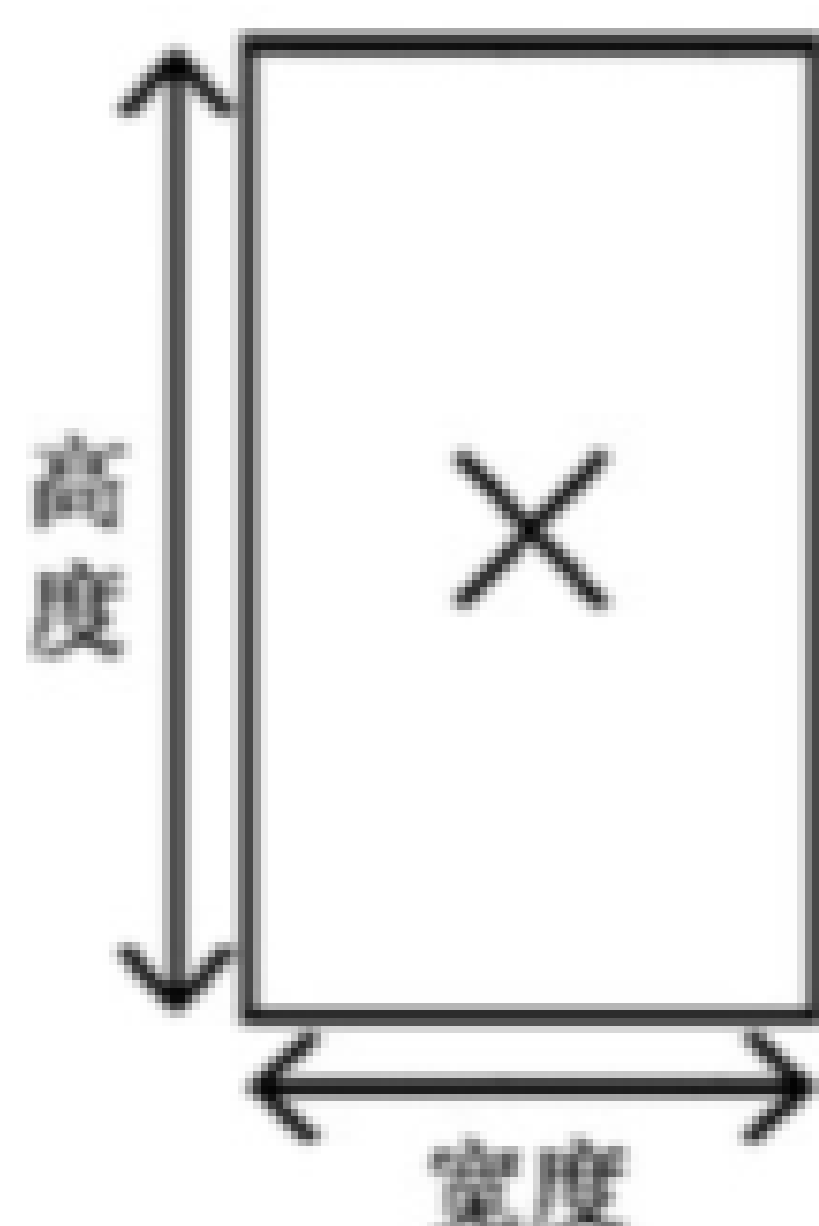


图 74 矩形

- c) id square width: 每个正方形都由宽度来定义, 定义应出现在一行上, 参考点在正方形的中心, 图 75 将参考点标记为 X。

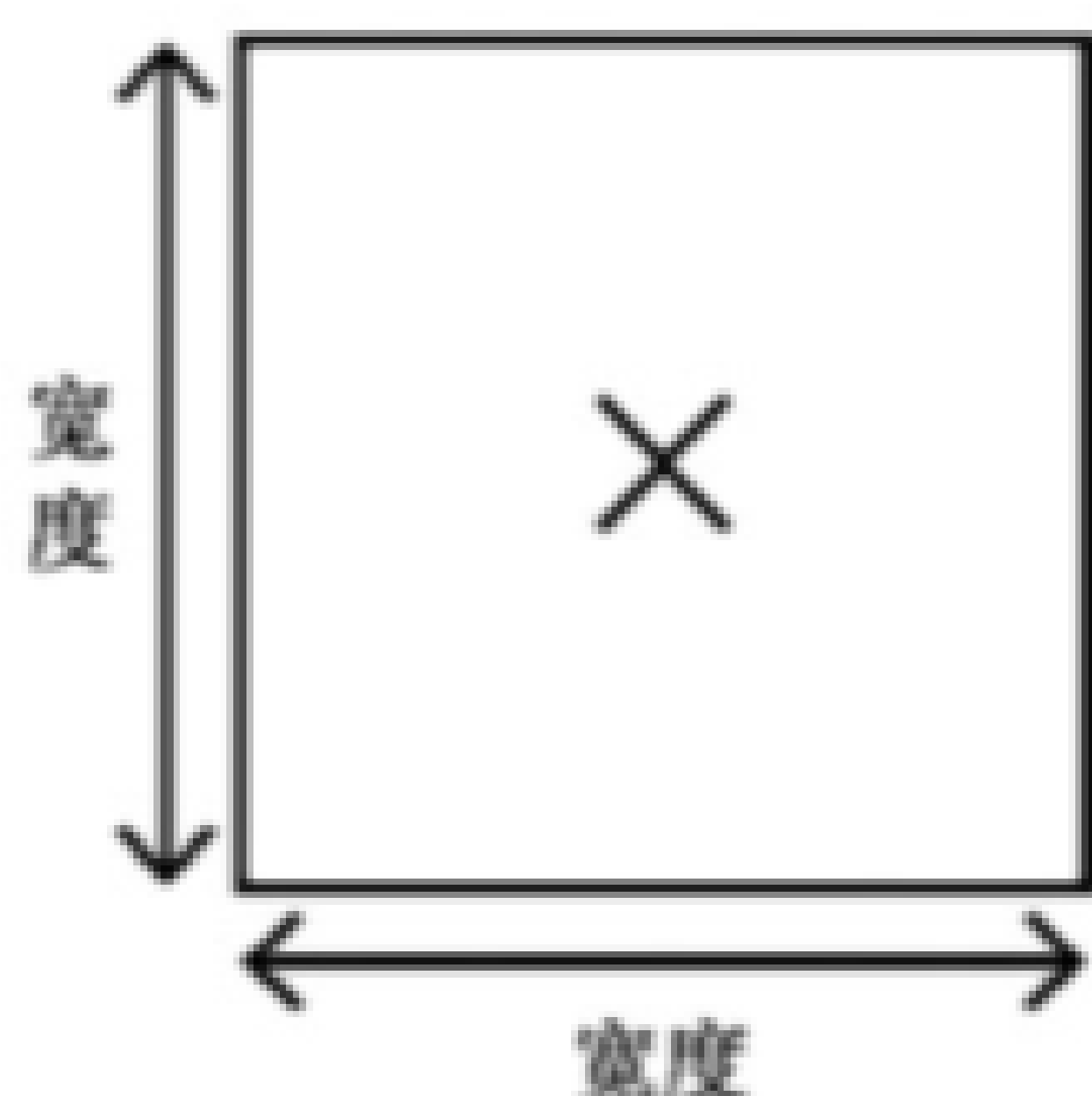


图 75 正方形

- d) Id circle diameter: 每个圆都是由直径来定义的, 定义应出现在一行上, 参考点在圆心, 图 76 将参考点标记为 X。

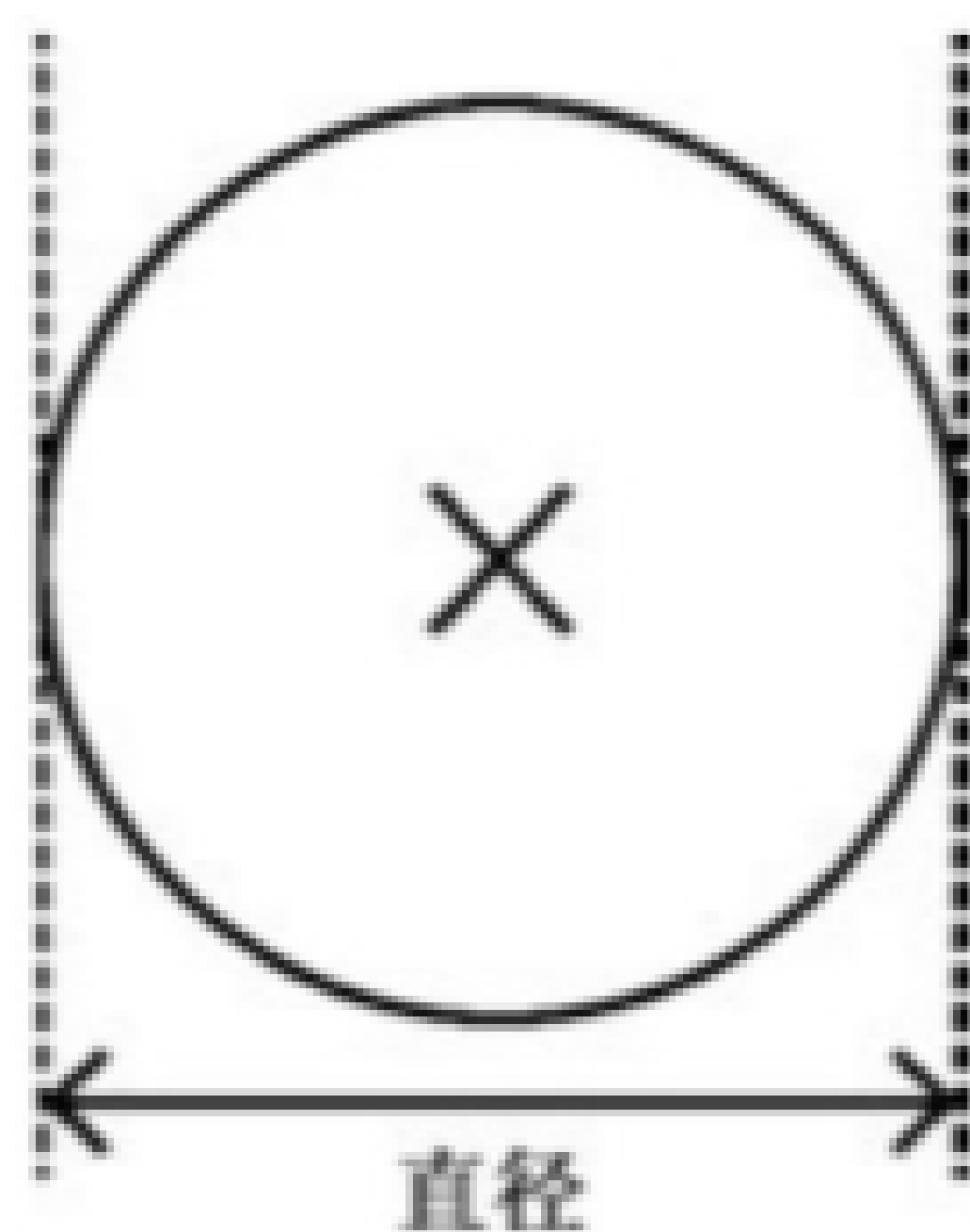


图 76 圆形

- e) id annular outerDiam innerDiam: 每个环由外径和内径定义, 定义应出现在一行上, 参考点位于环形中心, 图 77 将参考点标记为 X。

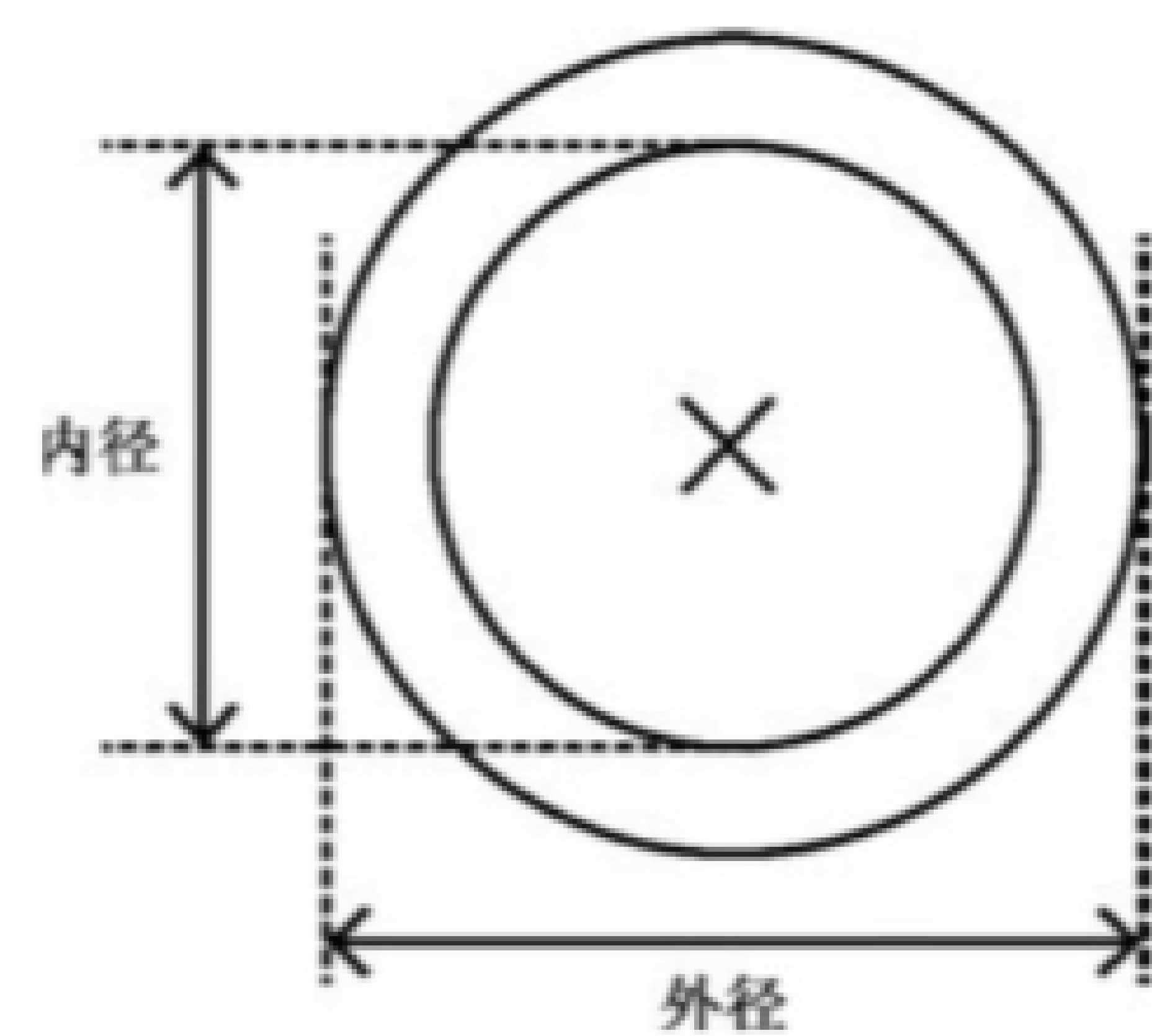


图 77 环形

- f) id oblong width left right: 每个长方形由宽度、左边和右边定义, 定义应出现在一行上, 参考点如图 78 所示, 标记为 X。

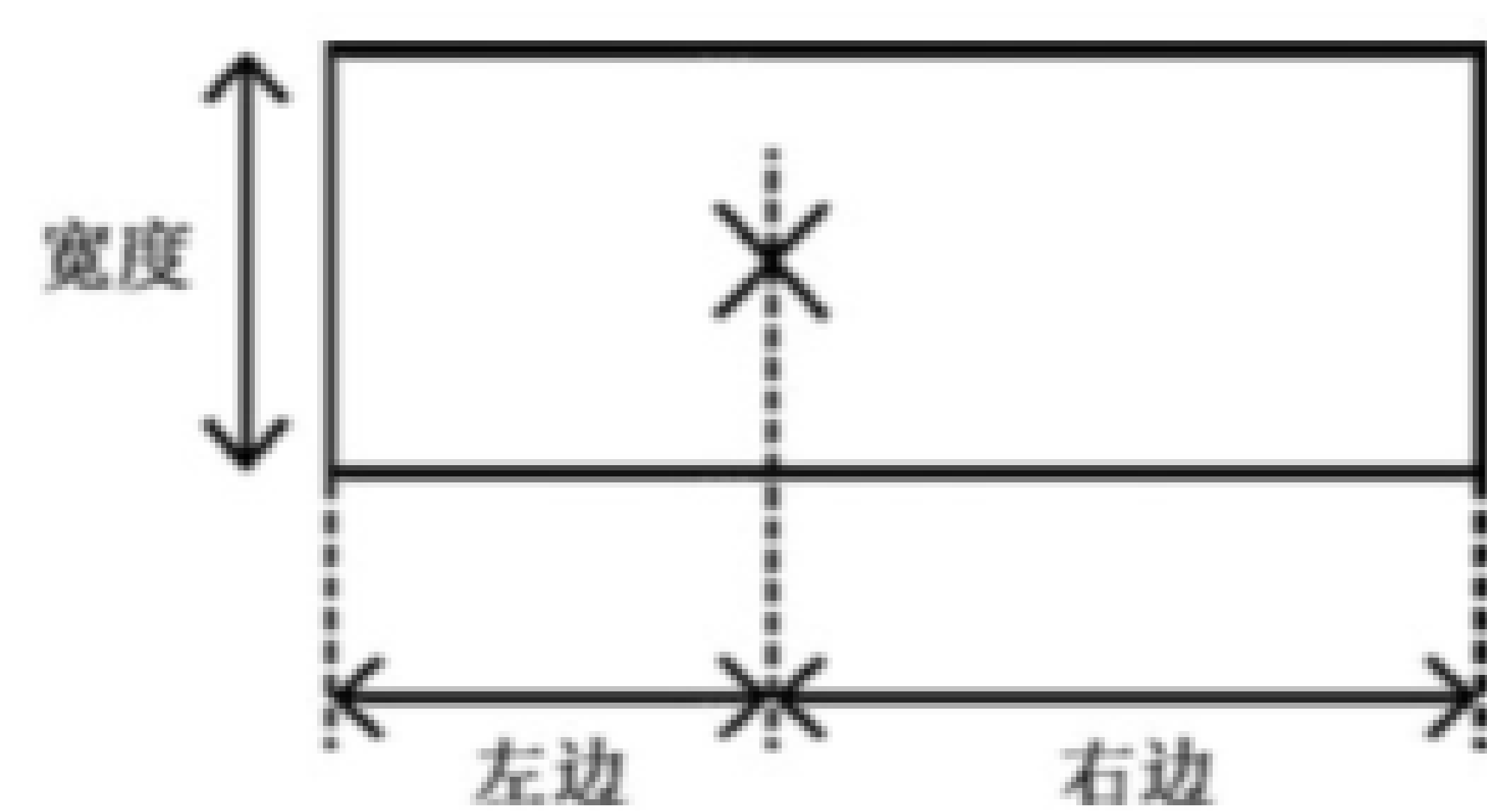


图 78 长方形

- g) id bullet widthleft right: 每个弹形都由宽度、左边和右边定义, 定义应出现在一行上, 参考点如图 79 所示, 标记为 X。

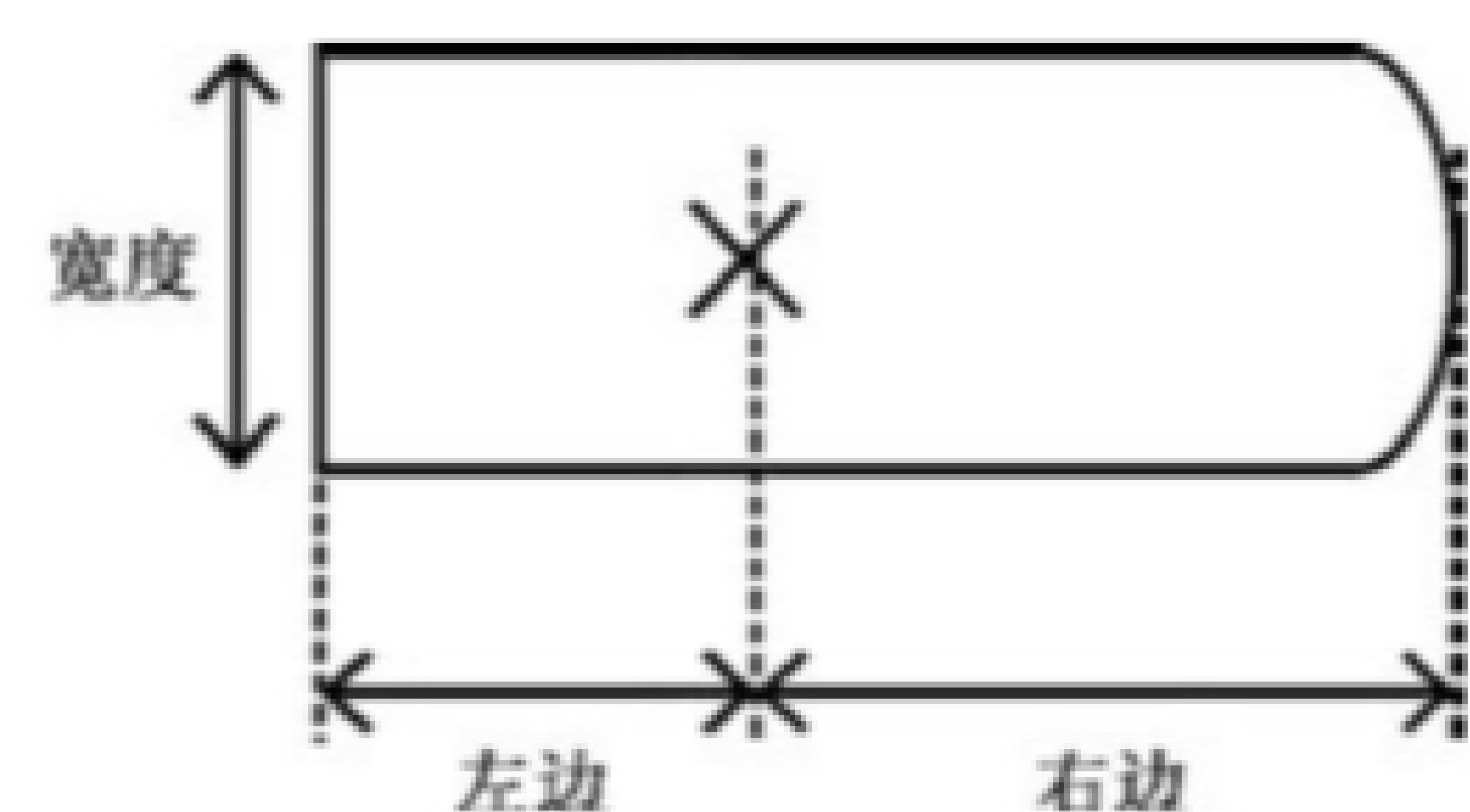


图 79 弹形

- h) id finger width left right: 每个指形由宽度、左边和右边定义, 定义应出现在一行上, 参考点如图 80 所示, 标记为 X。

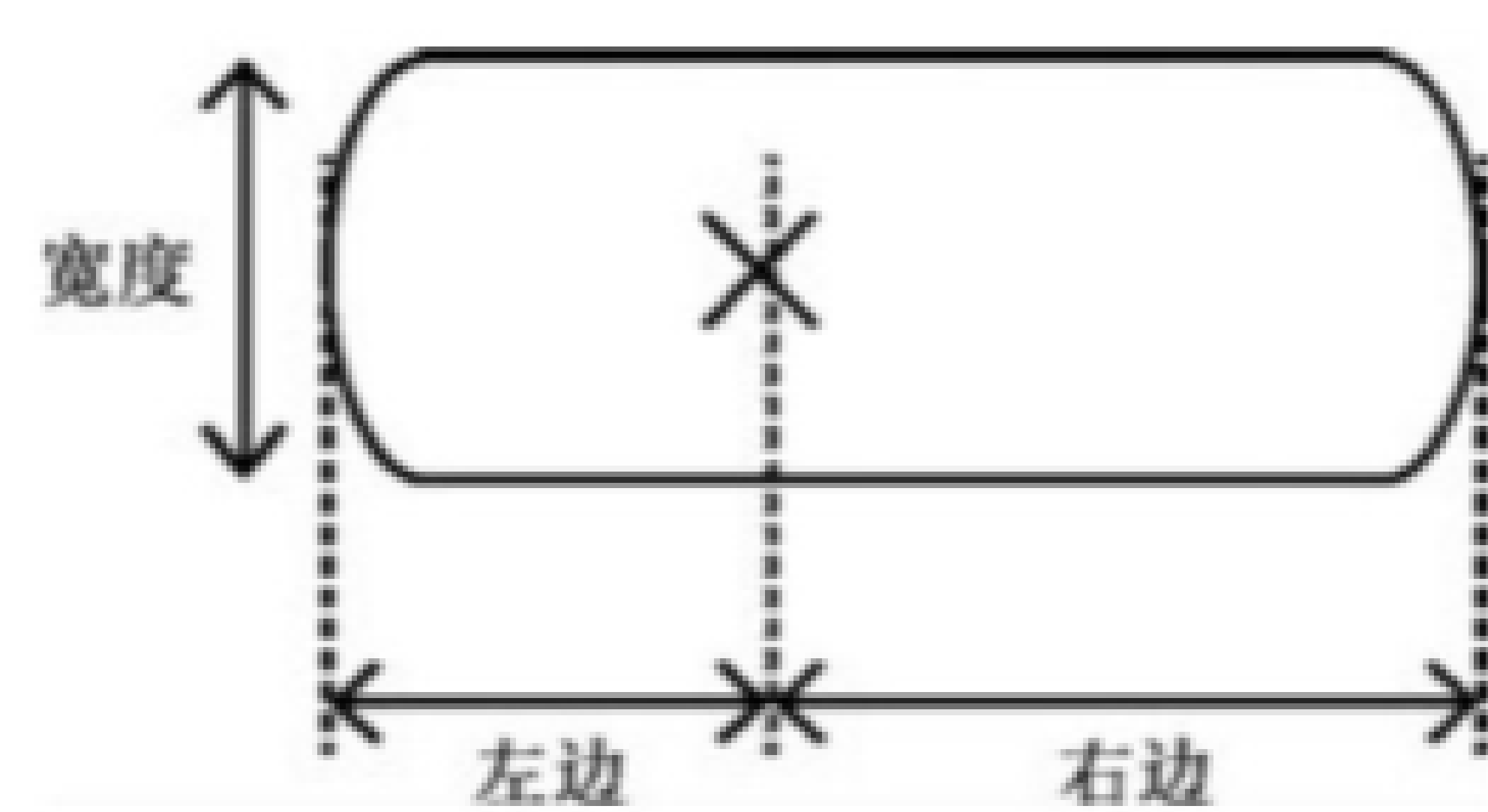


图 80 指形

- i) id composite { 1st\_point\_segment1 segment2... }: 复合形状由线和/或圆弧组成, 该定义由 {} 括起来, 可以在多行上显示, 有四种类型的线段: 直线、顺时针圆弧(弧)、逆时针圆弧(rarc 或反向圆弧)和由三点(圆弧 3)定义的圆弧。两个点之间没有关键字表示, 它们是通过一条直线连接的, 两个点之间的关键字 arc、rarc 或 arc3 分别表示它们通过顺时针弧、逆时针弧或三点方

法弧连接。对于顺时针圆弧和逆时针圆弧，在圆弧结束后出现一个圆弧原点。对于圆弧 3，在圆弧结束点之后出现一个中间点。如图 81 所示的复合形状可以描述为：

```
2 composite { xa ya xb yb xc yc rarc xd yd xp yp xe ye xf yf arc xg yg xq yq xh yh arc3 xi yi
xr yr }
```

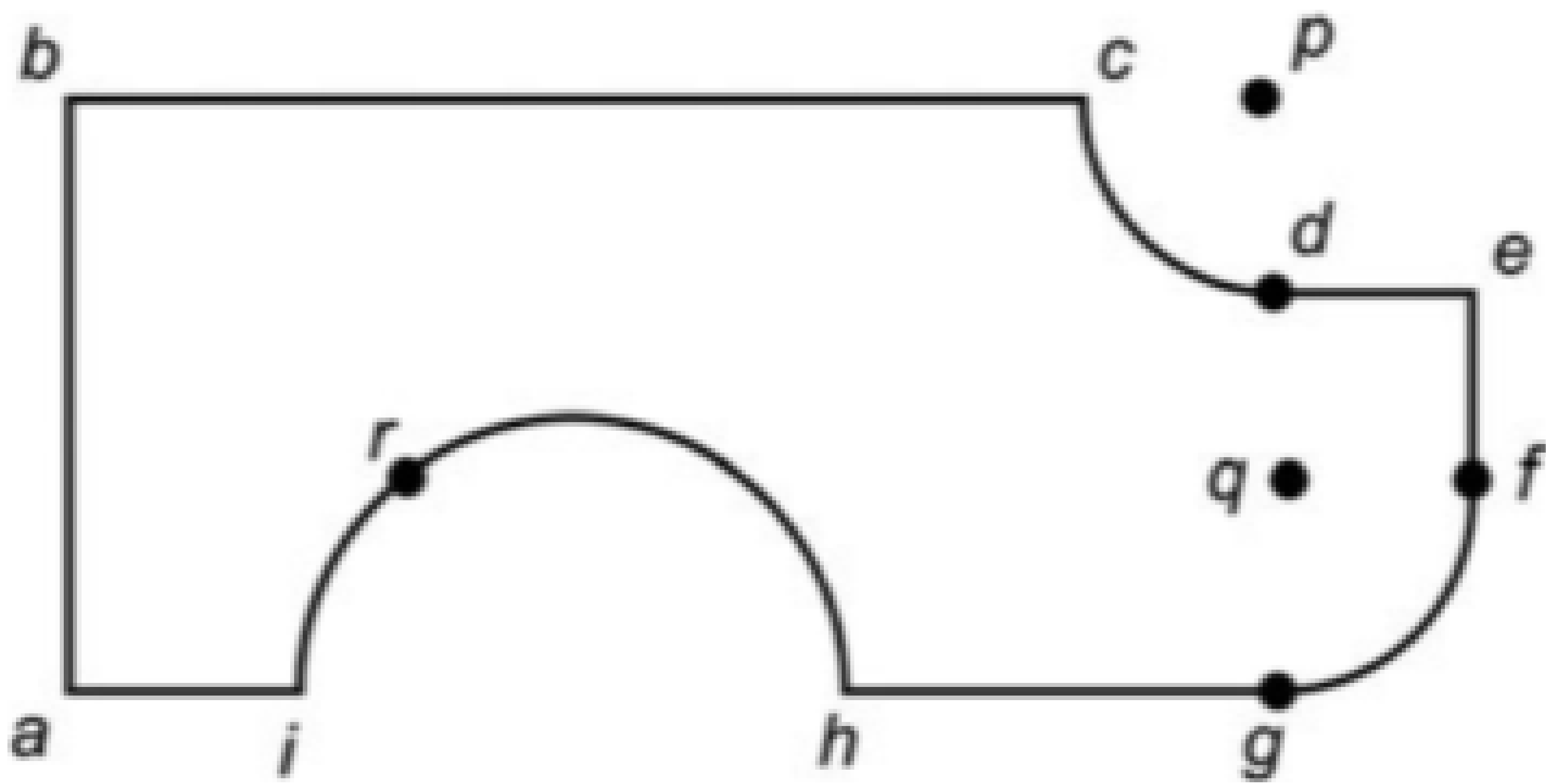


图 81 复合形状

其中开头的 2 是形状标识符，该形状由 a-b 和 b-c 直线构成，然后是从 c 到 d 的逆时针圆弧，在 p 处有原点，然后是 d-e 和 e-f 直线，然后是从 f 到 g 的顺时针圆弧，在 q 处有原点，然后，直线 Gh 和三点法从 h 到 i，在 r 处有一个中间点，最后，没有定义从 i 到 a 的直线，因为默认情况下，直线会将最后一点连接到第一个点。

同样的形状也可以描述为：

```
2 composite { xa ya xi yi arc3 xh yh xr yr xg yg rarc xf yf xq yq xe ye xd yd arc xc yc xp yp
xb yb }
```

11.6.2 例句

下面是.shape 部分的一个示例。

```
.shape
1 circle 50
2 circle 80
3 circle 300
4 rectangle 400 400
5 rectangle 700 5600
6 circle 750
7 composite {
275 -200
275 200
-275 200
-275 -200
}
8 composite {
400 -100
400 100
-400 100
```

```
—400 —100
}
.end shape
```

11.7 Board geometry 部分

11.7.1 概述

.board\_geom 部分由以下内容组成：

```
.board_geom
definition
.end [board_geom]
```

板或封装的外部几何形状由下列四种方法之一定义：

```
polygon { x1 y1 x2 y2... }
composite { 1st_point_segment1 segment2... }
shape shapeID x y rotation mirror
shape shapeID x y mirror rotation
```

定义多边形和复合形状的方法与.shape 部分(11.6)中描述的方法相同。

形状是通过定义形状原点(局部原点)的形状 ID、x 和 y 坐标(全局坐标)、逆时针旋转度数和镜像指示器来放置的。镜像指标如下：

- X:X 轴镜像；
- Y:Y 轴镜像；
- N:无镜像。

如果在旋转后出现镜像指示器,则在旋转之后执行镜像;如果镜像指示器出现在旋转之前,则在旋转之前执行镜像。

镜像和旋转操作都是针对形状原点(局部原点)执行的。  
通过使用以下一种或多种方法来定义孔(切口或空隙)。

```
void_polygon { x1 y1 x2 y2 ... }
void_rectangle width length x y
void_square width x y
void_circle diameter x y
void_composite { 1st_point_segment1 segment2 ... }
void_shape shapeID x y rotation mirror
void_shape shapeID x y mirror rotation
```

void\_polygon、void\_composite 和 void\_shape 以相同方式分别定义多边形、复合形状和形状。

void\_rectangle、void\_square 和 void\_circle 在 .shape 部分(11.6)中定义矩形、正方形和圆的方式相同,它们放置在坐标 x 和 y 处。



11.7.2 例句

下面是.board\_geom 部分的一个示例。

```
.board_geom
polygon {
    -50000 -40000
    50000 -40000
    50000 40000
    -50000 40000
}
.end board_geom
```

11.8 Padstack 部分

11.8.1 概述

.padstack 部分由以下内容组成：

```
.padstack
definition
.end [padstack]
```

本节定义焊盘堆栈，每个焊盘被定义如下：

```
padstackID { pad1 pad2... }
```

其中，padstackID 是.via 部分(11.13)中通孔引用的数字，该节按顺序从 1 编号，如下所述，焊盘仅通过使用.shape 部分中预定义的形状来定义：

```
signalLayerNum shapeID shapeROT [ apshapeID apshapeROT ]
```

- 其中：
- signalLayerNum :层部分中出现的信号层号。
  - shapeID :形状标识符。
  - shapeROT :形状的逆时针旋转角度，以度为单位。
  - apshapeID :反焊盘形状标识符，反焊盘定义是可选的。
  - apshapeROT:逆时针旋转角度的反形状，以度为单位。

11.8.2 例句

下面是.padstack 部分的示例。

```
.padstack
1 {
```

```

        1 3 0 4 0
        2 3 0 4 0
        3 3 0 4 0
        4 3 0 4 0
    }
    2 {
        1 3 0 4 0
        2 3 0 4 0
    }
    3 {
        2 3 0 4 0
    3 3 0 4 0
        4 3 0 4 0
    }
    4 {
        1 12 0
    }
    5 {
        1 13 0
    }
    6 {
        1 14 0
    }
    .end padstack
```

11.9 Part 部分

11.9.1 概述

.part 部分由以下部分组成：

```
.part
definition
.end [part]
```

本节描述了一个零件，每个部分定义如下：

```
partName [ shape llx lly urx ury height [type value [noflip ] [material ]]] {pin1 pin2...}
```

其中，partName 是部件和形状的名称，llx、lly、urx、ury、height 和 noflip 都是可选的。如果出现关键字 noflip，当该部分放置在该层下面时，将不会翻转该零件。

partName 部件的名称  
shape 该部件的顶部视图形状

R	矩形
C	圆形
D	圆角钻石

*llx, lly* 边界框的左下角坐标

*urx, ury* 边界框的右上角坐标

*height* 高度

*Type* 部件类型

R	电阻器
L	电感器
C	电容器
S	焊球
D	裸芯片
M	塑封料
O	其他类型

*value* 部件值( $m\Omega$  表示电阻,  $nH$  表示电感器,  $PF$  表示电容器), 对于所有其他部件类型, 将其设置为零, 如果这三个值在括号()中, 则部分值为电阻器、电感器和电容器; 例如(20.05.3.5)。

*material* 是可选的, 材料的名称用“”括起来。

引脚定义如下:

*pinName x y ioType [padstackID]*

其中:

*pinName* 引脚的名称, 如果不知道名称, 则使用顺序编号作为名称。

*x y* 引脚相对于本地原点的位置。

*ioType* 引脚 I/O 类型:

D 驱动器引脚

R 接收器引脚

B 双向引脚

DT 驱动器终端

RT 接收器终端

*padstackID* 可选, 它是焊盘标识符(如果 *pad* 堆栈是未知的, 则为 0)。

11.9.2 例句

下面是.Part 部分的一个示例。

```
CAP0603B R -300 -150 300 150 400 C {
    1 -425 0 B 23
    2 425 0 B 23
}
CAP1005B R -650 -250 650 250 400 C {
    1 -500 0 B 24
```

```
2 500 0 B 24
}
REGULATOR R -2000 -2000 2000 2000 0 D {
1 -1800 1050 B 21
10 1800 -350 B 21
11 1800 350 B 21
12 1800 1050 B 21
13 1050 1800 B 21
14 350 1800 B 21
15 -350 1800 B 21
16 -1050 1800 B 21
2 -1800 350 B 21
3 -1800 -350 B 21
4 -1800 -1050 B 21
5 -1050 -1800 B 21
6 -350 -1800 B 21
7 350 -1800 B 21
8 1050 -1800 B 21
9 1800 -1050 B 21
}
SMA_X1 R -2750 -2750 2750 2750 0 O {
1 0 0 B 18
2 -2000 2000 B 18
3 -2000 -2000 B 18
4 2000 -2000 B 18
5 2000 2000 B 18
}
.end part
```

11.10 Component 部分

11.10.1 通则

```
.component 由以下部分组成：
.component
definition
.end [component]
```

本节描述元器件的放置，每一行显示一个元器件位置，每个放置的元器件定义如下：

```
U-name partName x y layer rotation [stackComp] [(R L C 0)]
```

其中：  
U-nameU-name 也称为位置标识符或引用标识符，名字中不准许有空格。

*partName* 元器件的名字,名字用双引号(“”)括起来。

*x y* 元件原点相对于板原点的位置。

*z* 如果给定 *z* 和 *layer*,那么 *z* 表示在 +*n* 或 -*n* 指定的方向上与板层的距离。这将允许在主板上嵌入设备。

如果给定 *z* 和 *stackComp*,如果 *z* 为正,则 *z* 表示“*stackComp*”最上层的距离;或者如果 *z* 为负,则 *z* 表示与“堆叠 *comp*”的最底层的距离。

*layer* 放置层编号:

+*n* 用于上层

-*n* 用于下层

*rotation* 元器件相对于元器件原点(局部原点)的逆时针旋转。

*stackComp* 可选项,它是将元器件堆叠在其上的元器件名称,如果这样定义,则忽略层号。

*R L C 0* 可选项,它是元器件的 R、L、C 和括号中的其他值;例如,(50,10,250 0),单位是 mΩ, nH, 或 pF,第四个有效值是供将来使用。

11.10.2 例句

以下是 .component 部分的示例。

```
.component
C10 CAP0603B -8584.7 -4104.9 -4 0
C11 CAP0603B -8584.7 -6355.9 -4 0
REGULATOR REGULATOR -12598 10183.9 1 0
SMA_X1 SMA_X1 1570.7 32293.2 1 0
SMA_X2 SMA_X1 -7429.3 32293.2 1 0
.end component
```

11.11 Net attribute 部分

.netattr 部分由以下内容组成:

```
.netattr
definition
.end [netattr]
```

本节定义 net 属性,如 net 类别、允许延迟和其他用户属性。  
每个属性组定义如下:

```
id {attrributeName1=value1 attributeName2=value2 ...}
```

其中,*id* 是一个属性组号,它由 netlist 部分中的网引用,并按顺序从 1 编号,可以为一个组定义多个属性,每个属性组定义可以出现在多个行上,属性赋值出现在大括号{}中,每个属性赋值由 *attributeName*、等号(=)和 *value* 组成。

11.12 Netlist 部分

11.12.1 概述

.netlist 部分由以下内容组成:

```
.netlist
definition
.end [netlist]
```

本节定义 netlists,每个网络被定义如下:

```
netName netType attributeID {node1 node2 ...}
```

其中:

netName 网络名。  
netType 网络类型:  
S 用于信号网络(s 表示断开)  
P 用于电源网络(p 表示断开)  
G 用于接地网络(g 表示断开)  
attributeID 网络属性标识符,零表示属性未知。

每个节点定义如下:

```
U-name pinNumber ioType [{x y layer}]
```

其中:

U-name U-名称,减号(-)表示一个未知的 U-名称。  
pinNumber 引脚编号或引脚名称。  
ioType 引脚 I/O 类型:  
D 驱动器引脚  
R 接收器引脚  
B 双向引脚  
DT 驱动器终端  
RT 接收器终端  
如果使用部件为段中的引脚定义的类型,则应出现占位符减号(-)。

x y 是引脚原点相对于板原点的位置。

layer 是放置层的编号。

引脚位置可以出现在节点定义的末尾,它是可选的,应由{}括起来,将焊球的层数设置为负数。

11.12.2 例句

下面是.netlist 部分的示例。

```
.netlist
"FKOUT[2]" S 0 {
    FKB48 M1 B
    SMA_X4 1 B
}
"N1" S 0 {
    XTAL 3 B
```

```

        MCR1_R7 B B
        MCR1_R8 B B
        GRM1_C2 A B
    }
    "PCIREFCLK_N" S 0 {
        SOC AE12 B
        PCIE A14 B
    }
    "PCIREFCLK_P" S 0 {
        SOC AF12 B
        PCIE A13 B
    }
    "PCITR0_N" S 0 {
        SOC AE14 B
        PCIE A17 B
    }
}
.end netlist
```

11.13 Via 部分

11.13.1 概述

.via 由以下部分组成：

```

.via
definition
.end [via]
```

本节定义了导通孔，一行将显示一个导通孔定义，每个导通孔被定义如下：

```
viaName padstackID padstackROT shapeID shapeROT [ thickness ]
```

其中：

- viaName 通过路由部分中的路由引用的导通孔名称。
- padstackID 焊盘堆栈标识符，零表示导通孔不存在焊盘。
- padstackROT 焊盘堆栈的逆时针旋转角度。
- shapeID 导通孔的形状标识符。
- shapeROT 导通孔形状的逆时针旋转角。
- thickness 可选项，这是导通孔壁的厚度，如果厚度为 0，则用材料填充导通孔（见图 82）。

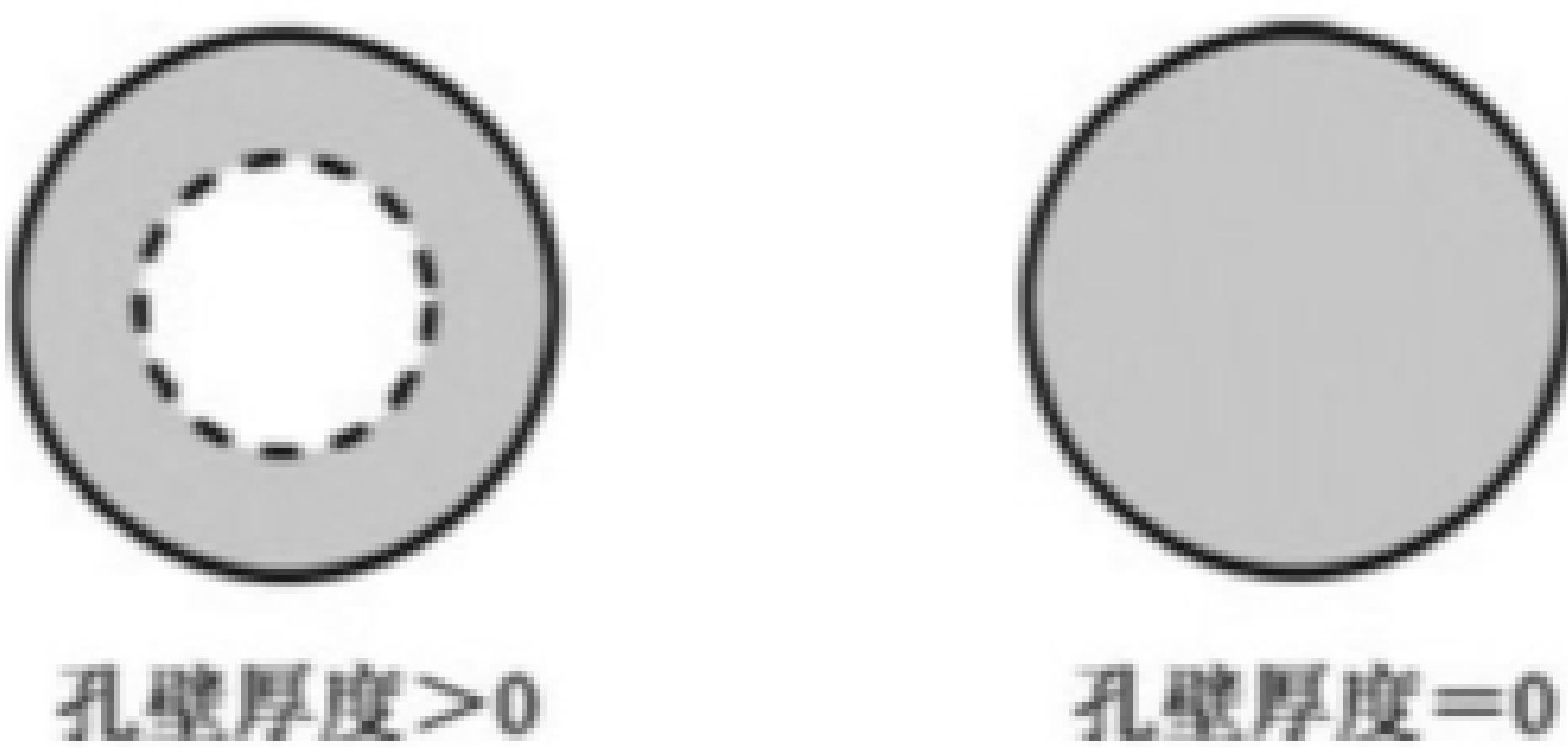


图 82 导通孔

11.13.2 例句

下面是一个.via 部分的示例。

```
.via
VIA14 1 0 2 0
VIA12 2 0 2 0
VIA24 3 0 2 0
VIA13 4 0 2 0
VIA14B 18 0 17 0
.end via
```

11.14 Bondwire 部分

11.14.1 概述

.bondwire 由以下部分组成：

```
.bondwire
definition
.end [bondwire]
```

本节定义了键合丝的几何形状，每一行定义一个键合丝，每条键合丝被定义如下：

```
id type material diameter hw hd alpha beta [ profile_name ]
```

其中，*id* 是通过路由部分中的路由引用的数字，并从 1 依次编号，如图 83 和图 84 所示，*type* 键合丝类型：

- D 用于自顶向下的配置
- U 用于倒装配置
- material* 材料名或电导率[1/(Ω·mm)]，如果它是一个名字，将用“”括起来。
- diameter* 键合丝直径。
- hw* 键合丝环的高度。
- hd* 裸芯片的高度：
- H<sub>die\_pad</sub>* 顶盖金属层高度；
- H<sub>die\_pad</sub>* 底盖金属层高度。
- alpha* 以度为单位的裸芯片侧角度。
- beta* 封装基板的边角，以度为单位。
- profile\_name* 与*id* 关联的键合丝外形的名称。



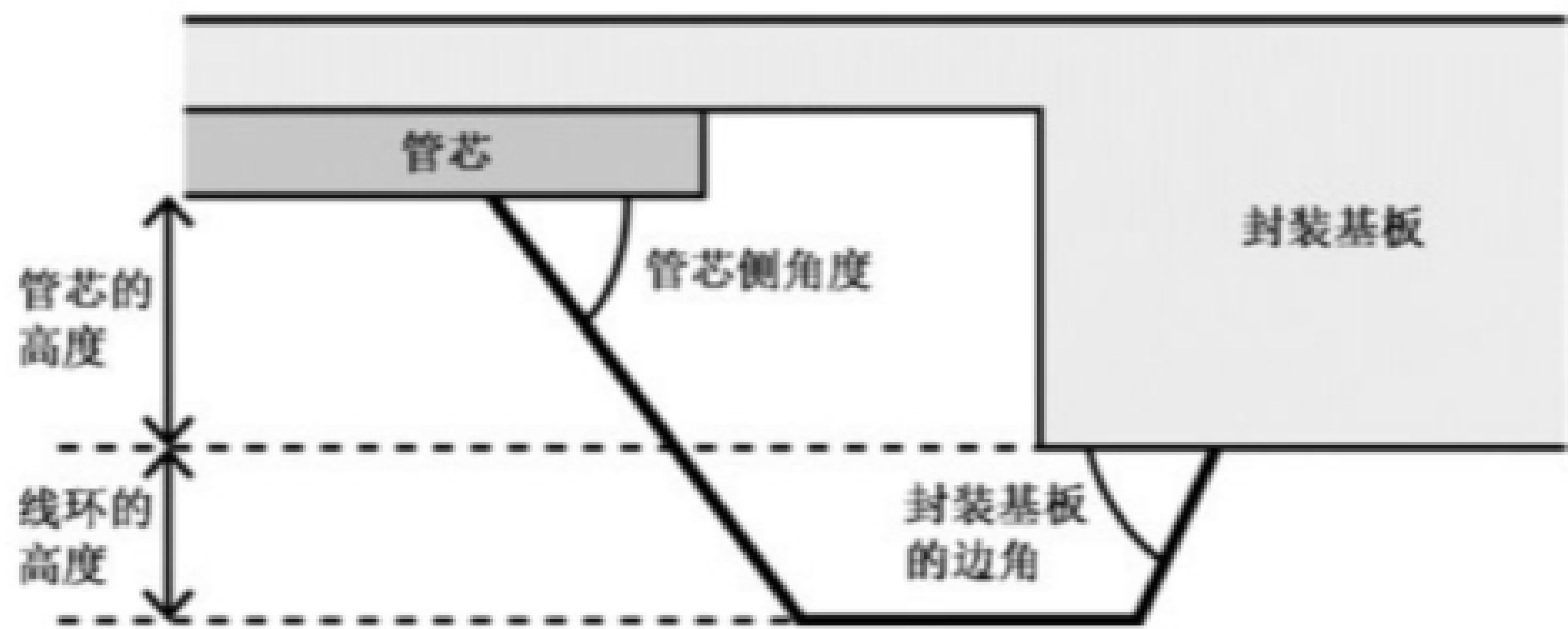


图 83 自顶向下配置

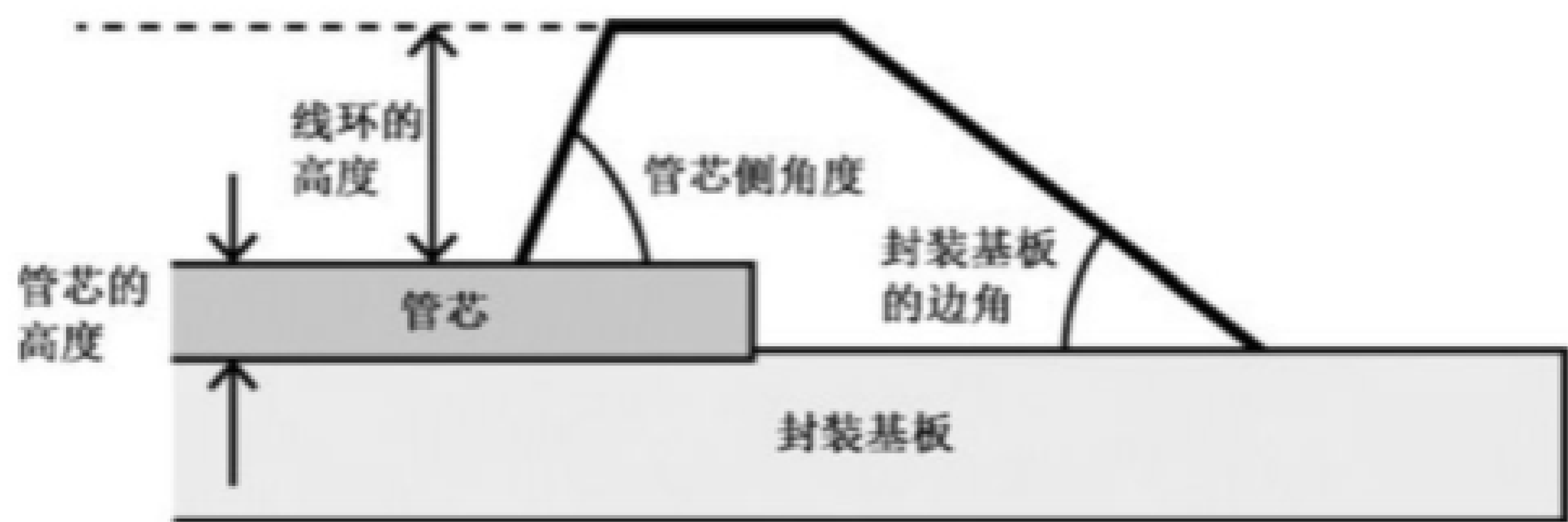


图 84 倒装配置

11.14.2 例句

下面是一个.bondwire 部分的示例。

```
.bondwire
1 U "GOLD" 20 70 290 60 20 innerwire
2 U "GOLD" 20 140 290 60 40 outerwire
.end bondwire
```

11.15 Route 部分

11.15.1 概述

.route 部分由以下内容组成：

```
.route
definition
.end [route]
```

本节定义外形，每个外形定义如下：

```
netName { segment1 segment2... }
```

其中，netName 是出现在.netlist 部分中的一个网络的名称，每个路径由一个或多个段组成，每个段由下列方法之一定义。所有这些方法都由以下方式定义：

```
segment_type signal_layer_number(s) segment_definition
path layer width { 1st_point_segment1 segment2... }
via beginLayer endLayer viaName x y rotation [ mirror ]
```

```
bondwire beginLyr endLyr bondwireID xb yb xe ye [ die1 die2 ]
polygon layer { x1 y1 x2 y2... }
rectangle layer width length x y
square layer width x y
circle layer diameter x y
annular layer outerDiem innerDiem x y
composite layer { 1st_point_segment1 segment2... }
shape layer shapeID x y rotation mirror
shape layer shapeID x y mirror rotation
void_polygon layer { x1 y1 x2 y2... }
void_rectangle layer width length x y
void_square layer width x y
void_circle layer diameter x y
void_composite layer { 1st_point_segment1 segment2... }
void_shape layer shapeID x y rotation mirror
void_shape layer shapeID x y mirror rotation
```

路径段的定义方式与在.shape 部分中定义复合形状的方式相同,只是路径段具有宽度,并且最后一点不会自动连接到第一个点。

导通孔段需要初始层和终端层号,而其他段只需要一个层号,导通孔部分还需要以下内容:

```
viaName 导通孔部分中定义的导通孔的名称
x y 导通孔位置
rotation      焊盘的逆时针旋转角,以度为单位
mirror        可选项,焊盘镜像标志:
                Y          焊盘堆栈镜像
                N          焊盘堆栈不镜像
```

键合丝段要求如下:

```
beginLyr 起始点的信号层数。
为了指示裸芯片,输入一个负线组标识符。
endLyr 结束点的信号层数。
为了指示裸芯片,输入一个负线组标识符。
bondwireID 键合丝标识符(0 表示未知)
xb yb 起点坐标
xe ye 终点坐标
die1 die2 可选裸芯片元件名称,只有当begin_lyr 和/或end_lyr 为负值时,名称才会出现在这里。
所有其他段类型的定义方式与在.shape 节或.board_geom 节中定义的方式相同。
```

11.15.2 例句

下面是.route 部分的一个示例。

```
"AGND" {
via 1 4 VIA14B -429.3 34293.2 0 N
via 1 4 VIA14B -429.3 30293.2 0 N
```

```
via 1 4 VIA14B -9429.3 34293.2 0 N
shape 1 7 -24821.9 24278.3 0 N
shape 1 12 -3779.3 23603.4 0 N
composite 2 {
  -47360 23040
  -15983.4 23040
  -15360 22416.6
  -15360 18416.6
  -14960 18016.6
  -14960 13616.6
  -14783.4 13440
  -4640 13440
  -4640 22416.6
  -4016.6 23040
  6560 23040
  6560 35360
  -47360 35360
}
void_composite 2 {
  -41295.6 25636.1
  arc -40975.6 25636.1 -41135.6 25636.1
  arc -41295.6 25636.1 -41135.6 25636.1
}
path 1 80 {
  -7825.9 27607.5
  -6985.3 27607.5
}
path 1 80 {
  -4082.8 25942.8
  -4082.8 25683
}
path 1 80 {
  -3719.3 23633.2
  -3380.3 23633.2
}
path 1 80 {
  -7390.9 22442.4
  -7130.9 22442.4
}
"AVDD33" {
  shape 1 8 -12948 11983.9 90 N
```

```

shape 1 7 -36821.9 23278.3 0 N
shape 1 7 -34821.9 23278.3 0 N
shape 1 7 -29821.9 24278.3 0 N
composite 3 {
-42160 23040
-15183.4 23040
-14560 22416.6
-14560 15840
-12640 15840
-12640 24960
-38816.6 24960
-39040 25183.4
-39040 25760
-42160 25760
}
via 1 3 VIA13 -13451.3 16882.6 0 N
via 1 3 VIA13 -36821.9 23278.3 0 N
via 1 3 VIA13 -24821.9 23278.3 0 N
shape 1 11 -13457.1 16897.2 270 N
shape 1 11 -41161.1 25621.3 0 N
via 1 4 VIA14 -41135.6 25636.1 0 N
}
"DDRAD[0]" {
shape 1 9 10900 0 0 N
shape 1 6 34806.5 -1623.2 0 N
shape 1 6 34547.6 17511.7 0 N
shape 1 14 26777.7 18269.6 0 N
via 1 4 VIA14 34200 2360 0 N
via 1 4 VIA14 34226.5 635 0 N
via 1 4 VIA14 33967.6 19939 0 N
path 1 80 {
33967.6 19939
33967.6 19661.7
33967.6 19508.4
33977.6 19498.4
33977.6 18875.6
34057.6 18795.6
34057.6 18577.3
34137.6 18497.3
34137.6 18279.1
34187.6 18229.1
34187.6 18162.6

```

34547.6 17802.6  
34547.6 17511.7  
}  
path 4 80 {  
34200 2360  
34200 2904  
34170 2934  
34170 11978  
34157.6 11990.4  
34157.6 12769.6  
33967.6 12959.6  
33967.6 19939  
}  
path 1 80 {  
34226.5 635  
34226.5 526.8  
34226.5 373.5  
34236.5 363.5  
34236.5 -259.3  
34316.5 -339.3  
34316.5 -557.5  
34396.5 -637.5  
34396.5 -855.8  
34446.5 -905.8  
34446.5 -972.3  
34806.5 -1332.3  
34806.5 -1623.2  
}  
path 4 80 {  
34200 2360  
34200 841.5  
34226.5 815  
34226.5 635  
}  
path 1 80 {  
34200 2360  
33650.8 2360  
33568.5 2442.3  
16116 2442.3  
14003.7 330  
13900 330  
13870 360

```
11750.9 360
11390.9 0
10900 0
}
}
.end route
```

---









中 华 人 民 共 和 国  
国 家 标 准  
大规模集成电路(LSI) 封装  
印制电路板共通设计结构  
GB/T 43863—2024

\*

中国标准出版社出版发行  
北京市朝阳区和平里西街甲2号(100029)  
北京市西城区三里河北街16号(100045)

网址:www.spc.net.cn

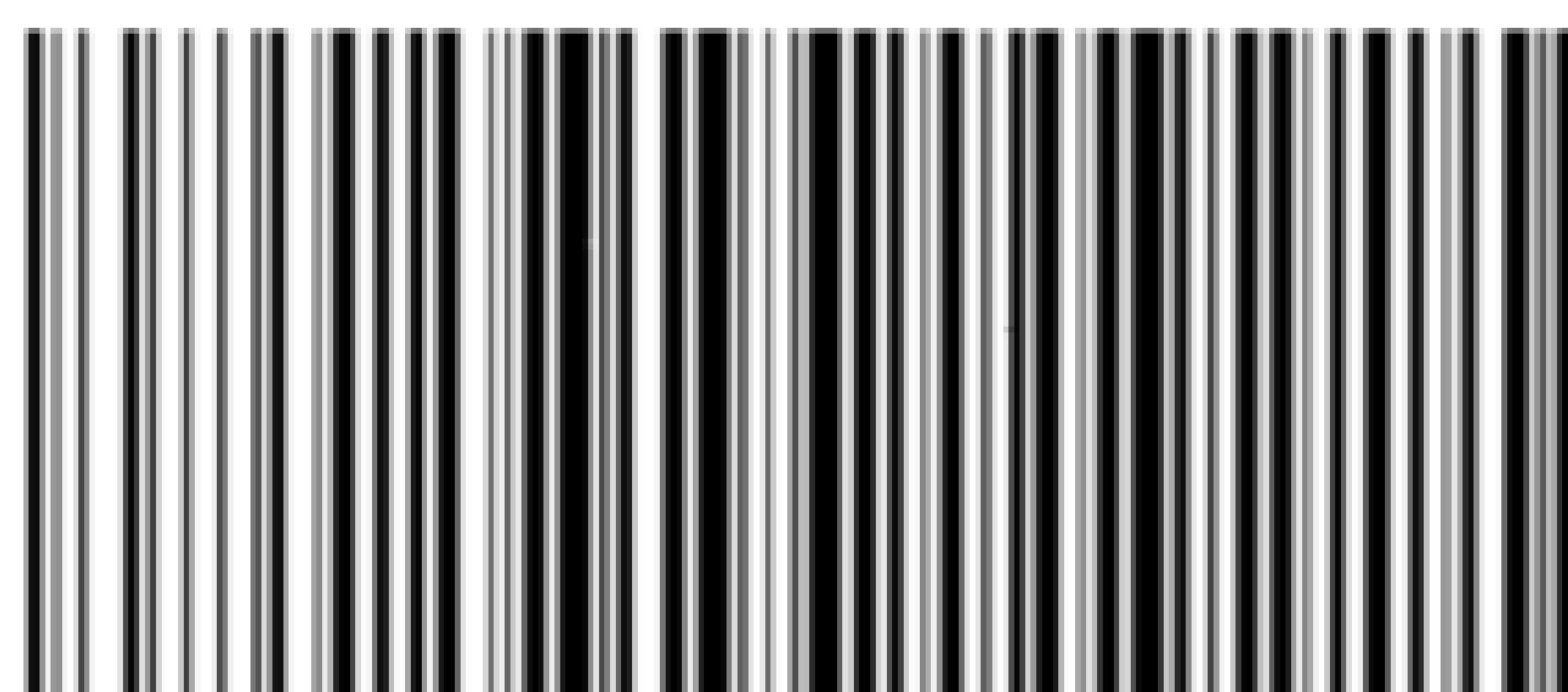
服务热线:400-168-0010

2024年4月第一版

\*

书号:155066·1-75560

版权专有 侵权必究



GB/T 43863-2024