

用 SKILL 语言实现 PCB 的原理图还原

陈 烨, 马鸣锦, 郭慧芳

(解放军信息工程大学 信息工程学院, 河南 郑州 450002)

摘 要: 目前市场上的 EDA 工具均只支持由原理图到网表的正向设计。介绍了自行开发的一种原理图还原系统, 作用是根据原理图的网表, 自动还原出 PCB 的原理图。具体是通过 SKILL 语言在 EDA 工具 Concept HDL 平台上二次开发实现的。

关键词: PCB; 原理图还原; Concept HDL; SKILL 语言; 消息传递系统

中图分类号: TP391

文献标识码: A

文章编号: 1000-7180(2007)04-0089-03

A Technology of PCB Schematic Recovery Based on SKILL

CHEN Ye, MA Ming-jin, GUO Hui-fang

(Institute of Information Engineering, PLA Information Engineering University, Zhengzhou 450002, China)

Abstract: This article discusses a software about the logic layout in the PCB inverted design system which is based on the Concept HDL that is one product of Cadence. The software that is written in SKILL can support the positive and inverted design.

Key words: PCB; the reversion of schematic; Concept; HDL; SKILL; MPS

1 引言

PCB 的逆向设计在许多方面有着广泛的应用。在对未知电路板的分析中, 通过逆向设计还原出其原理图, 便于分析电路设计原理^[1,2]。但是由于设计侧重点的不同, 目前市售的 PCB 设计工具均只支持正向设计, 即按照以下流程设计 PCB 板: 设计者绘制原理图 生成网表 物理元件加载 绘制电路板图。而 PCB 的逆向设计, 即由电路板分析还原出其电路原理图, 现有的 PCB 工具均不支持^[3,4]。因此, 研制支持 PCB 逆向设计的工具平台, 提高电路板解剖分析的工作效率, 已成为急需解决的问题。

如图 1 所示, 电路板逆向设计的流程为测量出电路连接关系, 得出测量网表文件, 最后根据测量网表还原出电路板的电原理图, 其中阴影部分为文中介绍的原理图还原系统实现的功能。

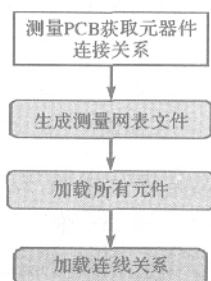


图1 电路板的逆向设计流程

2 实现方案

按照实现目标, 原理图逆向还原软件系统应具有以下功能:

(1) 将测量网表文件中的元件转换成逻辑元件符号, 并加载到设计界面的相应位置, 实现自动逻辑布局;

(2) 自动绘制电路连接关系;

(3) 逻辑元件库应具有可扩展性, 对库中没有的元件支持用户编辑新的元件符号并添加到逻辑元件库中;

(4) 提供对逻辑元件符号进行操作的各种常用工具, 比如常用形状对象、连线对象、符号对象、旋转、移动、选定等功能, 以支持用户对生成的原理图修改和整理。

针对逆向还原系统的设计要求, 有两种实现方案: 一是采用面向对象的编程方法, 自行设计操作界面, 对测量网表的解析、逻辑元件符号库的建立, 将测量网表中的元件转换为逻辑符号并加载到设计界面中, 以及一些常用的操作元件符号的工具, 都自行开发。采用这种方案, 风险很小, 虽然编程量很大, 但是几乎一定会实现。但缺点也是显而易见的, 自行开发的界面和常用工具, 肯定不能与专业 EDA 工具相媲美。另一种方案是对通用 EDA 平台进行改造, 二次开发逆向还原软件并融入其中, 这

收稿日期: 2006-04-17

种方案的优点是可以利用 EDA 平台丰富的绘图工具和庞大的逻辑元件库,提升原理图还原的质量与速度。但需要深入分析 EDA 工具内部的信息表示形式和开发扩展接口,并且没有先例可循,难度很大。

本着先进、可靠、可扩展的开发原则,选择了第二种开发方案,把首要解决的问题确定为选择适宜的 EDA 软件,对该软件进行改造,二次开发逆向还原软件。

Cadence 是市场上的主流 PCB 设计工具,它提供了二次开发语言 SKILL,允许设计者编制程序,创建自定义的命令,SKILL 语言为逆向还原软件提供了开发接口^[9]。Cadence EDA 中的 Concept HDL 作为原理图编辑器,提供了各种常用的绘图工具,方便用户修改原理图,Concept HDL 的逻辑元件库包含有各种常用的逻辑元件和电气元件,并支持用户制作添加新的逻辑元件,以此为平台绘制的原理图可以导出通用 EDIF (Electronic Design Interchange Format) 格式的网表文件与其它 EDA 工具进行数据交换。因此可以选择 Concept HDL 作为开发平台。

3 软件开发基础

3.1 SKILL 语言和 Concept HDL 的通信

SKILL 程序在 SKILL 解释器中运行,但是该解释器并没有和 Concept HDL 集成在一起,它们是通过 MPS 进行通信的。通过 MPS,SKILL 程序可以访问 Concept HDL 环境以及其中的对象,比如元器件、连线及其属性等。

MPS 以客户/服务端(client/host)的模型实现,提供远程进程调用机制。一个 MPS 服务就是一系列 API (Applications Procedural Interface, 应用程序接口) 的集合,这些 API 由服务端进程以远程函数的方式提供,客户端进程调用执行这些 API,如图 2 所示。

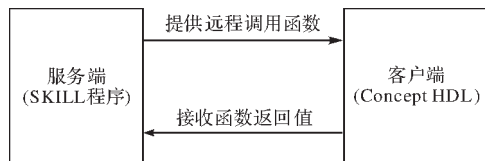


图2 服务端与客户端的关系

原理图逆向还原软件就是要研究如何使用 SKILL 提供的 API 接口来完成 Concept HDL 不能完成的某些任务,比如在 MPS 中连续向 Concept HDL 发命令加载元件和网络名,实现电路原理图的自动

生成。

3.2 SKILL 程序和其它进程的接口

SKILL 语言只能在 Cadence EDA 的环境中使用,应用范围很窄,所以功能有一定的局限性,比如文件的读写效率很低,对界面的编程能力不强等。为了增强逆向还原软件的开发功能,就需要研究 SKILL 程序与其它进程的通信接口,SKILL 程序与其它进程的通信是通过 IPC (Inter-Process Communication, 子进程通信) 机制实现的。

通过 IPC (Interprocess Communication) 机制,在 skill 程序中可以创建子进程并与之通信,子进程中可以运行任何可执行程序。IPC 机制提供了 skill 程序与其它可执行程序之间通信的渠道。

具体地说,通过 IPC 机制,用户的 skill 应用程序可以:

- 封装工具程序和实用程序;
- 通过标准 IO 通道与被封装的程序进行通信;
- 通过发送命令来控制被封装的程序;
- 允许被封装的程序执行在父进程中的 skill 命令;
- 在远程主机上执行子进程。

例如,SKILL 对文件读写的效率很低,且界面编程功能不强,相关问题需要和其它应用程序通信解决,SKILL 可以直接调用 .exe 文件执行,以下为调用记事本打开测量网表的代码:

```
n=strcat( cdsGetInstPath( ) "\\ notepad.exe " "w.txt ")
ipcBeginProcess( n)
```

对于需要用户交互的问题,如选择网表路径等,可以用 VC 开发,SKILL 可以直接调用 VC 的 .exe 执行文件:

```
m=strcat( cdsGetInstPath( ) "\\tools\\fet\\skill - 01.00\\con  
text\\Readfile.exe")
cid =ipcBeginProcess( m)
```

当程序执行至上面语句时,VC 的执行文件开始执行。

4 具体实现

4.1 元件加载

对测量网表文件进行读取,提取出元件属性和电路连接关系信息,这部分使用 VC++ 实现的,在 SKILL 程序中通过 IPC 机制调用。SKILL 程序在 MPS 机制中与 Concept HDL 通信,通过不断发命令实现原理图的正确还原。

以下是自动加载元件 SKILL 主要代码:

```
cid=ipcBeginProcess( strcat(cdsGetInstPath() "\\Readfile.exe"))
```

//SKILL 对文件读写的效率很低,且界面编程功能不强,因此加载网表的可执行文件 Readfile.exe 是由 VC 开发的,再由 SKILL 进行调用。

```
inport=infile(ipcReadProcess(cid 100))
when(inport
    while(gets(h inport) li=append1(li h))
//将加载网表中的信息读进一个组
ms=strcat("add" "" comp " " cnCreatePointString(1xy))
res=cnSendCommand( Concept HDLHandle ms
    ?runOptions (cnSetRunOptions ?display cnOUT_VERBOSE ?result cnOUT_VERBOSE))
```

//将元件添加至原理图上

```
cms=strcat ("signame" " " "unet " "cnCreatePointString
(pinxy))
cnSendCommand( Concept HDLHandle cms
    ?runOptions (cnSetRunOptions ?display cnOUT_VERBOSE ?result cnOUT_VERBOSE))
```

//将引脚的网络号加在引脚上

在生成的初始原理图上,同一网络中的引脚是以标识的同样的网络名表示的,自动布线可将网络名删掉,将同一网络中的引脚布线。自动布线充分利用了 Concept HDL 自带的连线(ROUTE)功能。

4.2 自动连线

以下是自动布线的 SKILL 程序的一些主要代码:

```
pro =cnGetProps (cnSendCommand (conceptHandle "group" " " "all"
?runOptions (cnSetRunOptions ?display cnOUT_VERBOSE ?result cnOUT_VERBOSE)))
```

//得到原理图上的所有的属性(Property)的信息,网络名包含在属性中

```
if( strcmp( unet "NET" 3) ==0
then
    li=append1(li unet)
```

//筛选出所有的网络名的属性

```
ms=strcat("route" " " cnCreatePointString(line1[m]) " "
cnCreatePointString( line1[m+1]))
rea=cnSendCommand( conceptHandle ms
    ?runOptions (cnSetRunOptions ?display cnOUT_VERBOSE ?result cnOUT_VERBOSE))
```

//将在同一网络中的引脚顺次连线

5 实例介绍

Concept HDL 提供了用户界面定制功能,可以增加新的菜单项。只需把新增的菜单命令关联到执行新命令的脚本文件,就可以将新增的功能集成到原有的应用中,实现统一的使用界面,增加了使用的方便性。图 3 为新增的菜单项。

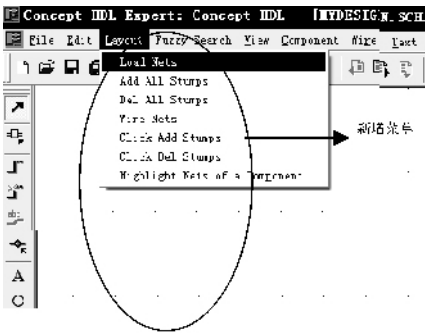


图3 Concept新增加的菜单

图 4 为逆向还原软件生成的整体原理图的一部分。为了更清晰的表现电路连接关系,引脚的连接关系使用标识在引脚坐标上的网络名来区别的。

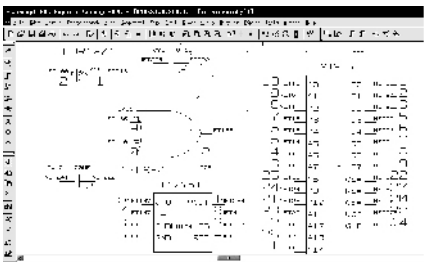


图4 原理图还原后的一部分

按照电路连接关系对原理图分页后,使用自动连线命令绘制原理图的连线,图 5 为绘制连线后的效果。少数自动连线程序走不通的连接关系是用虚线表示的,需要用户手工添加。

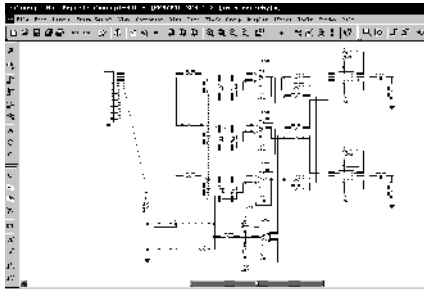


图5 自动连线

6 结束语

文中探讨了使用 SKILL 二次开发 Cadence (下转第 94 页)

适应度, P_m 和 P_c 的自适应调整与算法的收敛程度成反向, 从而有效地预防了算法收敛于局部最优; 并使好的进化结果得以保存。 k_1, k_2, k_3, k_4 取 $(0, 1)$ 区间的值。

3.4 算法步骤

(1) 设定种群规模为 size, 初始温度 T_0 , 衰减系数为 a , 设自适应交叉和变异系数分别为 k_1, k_2, k_3, k_4 。最优调度方案为 best_schedule, 最优适应度目标值为 best_result。令 $k=0, t_k=T_0$, 根据染色体编码规则产生 size 个可行解来形成初始种群 pop(k);

(2) 计算并评价当前种群适应度函数值 $f(i)$, 找出使函数 $f_i(t_k)$ 最小的染色体 i 和这个函数值 f 。如果 $k=0$, best_schedule= i , best_result= f ; 如果 $k \neq 0$, 但 $f < \text{best_result}$, 则 best_schedule= i , best_result= f ; 其中 $f_i(t_k)$ 为染色体 i 在温度 t_k 时的适应度函数值, $i \in \text{pop}(k)$;

(3) 判断是否满足停止条件, 若满足则停止, 输出最优调度方案和最优目标函数值; 否则, $t_{k+1}=at_k$, $k=k+1$, 按照自适应变异概率 P_m 和自适应重组概率 P_c 使用单点交叉对 pop(k) 中的每一条染色体产生新的染色体 j , 按照 Metropolis 接受准则中的接受概率接受或拒绝新染色体 j , 其接受概率为

$$P_i = \begin{cases} 1, & f_j(t_k) \leq f_i(t_k) \\ \exp\{-(f_j(t_k) - f_i(t_k))/T_k\}, & f_j(t_k) > f_i(t_k) \end{cases}$$

(4) 由此产生新种群 pop(k), 返回(2)。

经过上面的过程以后, 得到了针对协作型任务的最优调度方案, 按照 AOV 网中的制约关系依次在相应的资源上执行分配到该资源上的子任务。这样就可以在最大程度的缩短任务的执行时间。

4 结束语

文中对网格中最常见的协作型任务进行了分

(上接第 91 页)

EDA 系统的相关技术, 利用 SKILL 编程实现了从 PCB 到原理图的逆向还原系统。实践证明, 该系统能够快速准确还原出 PCB 的初始原理图, 大大提高了工作效率。

参考文献:

- [1] Cadence Design System. SKILL Language Reference, 1998
- [2] Cadence Design System, Cadence application infrastructure, 1998
- [3] 郭慧芳. 用 SKILL 语言对 Concept HDL 进行二次开发[J]. 电子设计应用, 2005, (3)

析, 采用 AOV 网来表示该类任务中各种子任务之间的关系, 同时针对该类任务提出一种基于遗传模拟退火算法的网格资源调度算法, 通过对染色体编码的设计来满足协作型任务中子任务之间的制约关系, 给出了该算法的原理和详细步骤, 该算法克服了遗传算法和模拟退火算法的各自缺点, 可以将网格中的资源进行优化分配。

参考文献:

- [1] HE X S, SUN X H, Gregor von Laszewski. QoS Guide Min-Min Heuristic for Grid Task Scheduling [EB/OL]. http://www.cs.iit.edu/~scs/pfiles/jcst_XHe-5-28.pdf. 2003
- [2] Ajith Abraham, Rajkumar Buyya. Nature's heuristics for scheduling jobs on computational grids[A]. The 8th Int'l Conf on Advanced Computing, Communications (ADCOM 2000) [C], Cochin, India, 2000.
- [3] Vincenzo Di Martino. Scheduling in a grid computing environment using genetic algorithms [A]. Marco Mililotti the 16th Int'l Parallel and Distributed Processing Symp (IPDPS2002) [C], Florida, USA, 2002
- [4] R Buyya, D Abramson, J Giddy. Economic Models for Resource Management and Scheduling in Grid Computing [J]. The Journal of Concurrency and Computation: Practice and Experience (CCPE), 2002, 14: 1507~1542.
- [5] Network Weather Service [EB/OL]. <http://nms.cs.utk.edu>

作者简介:

刘 岚 女, 硕士研究生。

苗 露 女。

- [4] 王中伟, 张盛兵, 沈戈, 赵勇. 一种在电路 SOC 验证接口设计方法研究[J]. 微电子学与计算机, 2005, 22(9): 68~70
- [5] 张松, 董玲, 于宗光, 等. 一种适合于 SoC 集成的 UART 核的设计实现[J]. 微电子学与计算机, 2005, 22(9): 10~13

作者简介:

陈 烨 女, 硕士研究生。研究方向为计算机接口技术。

马鸣锦 女, 教授, 硕士生导师。

郭慧芳 女, 讲师。