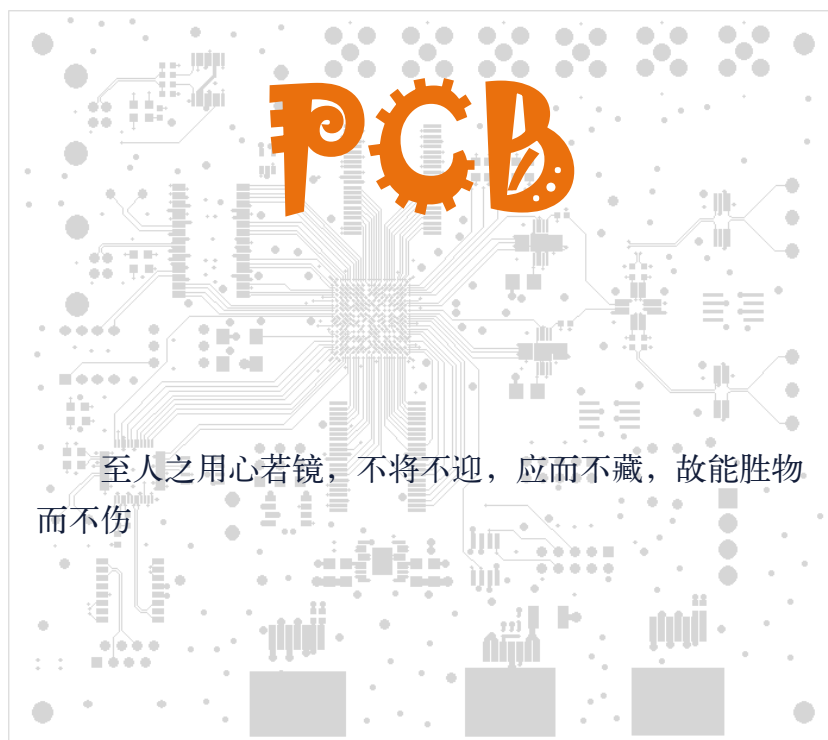


---

# PCB Design Tutorial

## PCB 设计 教程

---



---

整理：Wang Yu

整理时间：May 10, 2018

Email: [wyu0831@hotmail.com](mailto:wyu0831@hotmail.com)

---

Version: 1.0

# 目 录



1	使用 Capture 画原理图	2
2	建议规范	3
2.1	Allegro 配色方案	3
2.1.1	ETCH	3
2.1.2	Net	3
2.1.3	其他辅助层	3
3	使用 Allegro 画 PCB	4
4	平时的一些经验	5
4.1	制作 LOGO	5
4.2	Skill 教程	6
4.2.1	Allegro skill 介绍	6
4.2.2	Allegro skill 设置方法	6
4.2.3	AlignTool	7
4.3	制作异形过孔	7
4.4	从之前的工程中导入光绘设置	8
4.5	新建 SubClass	9
4.6	Allegro 设置快捷键	10
4.6.1	快捷键介绍	10
4.6.2	修改快捷键	10
4.7	钽电容画法	11
4.7.1	电源去耦电容的作用	11
4.7.2	钽电容画法	11
4.8	盲埋孔设计	12

## 说明



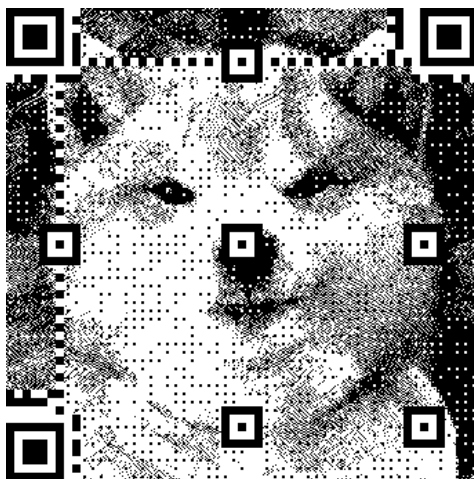
本文档画原理图和 PCB 教程主要来源于[Hu Jiadong](#)整理的文字版，内容来自于于博士视频。

技巧方面的内容是平时使用 Allegro 的技巧，做为教程的补充

规范部分来源于平时经验以及《高速数字设计课程内容》

教程发布在[著名同性交友网站-Github](#)上，地址是<https://github.com/wyu0725/PCBDesignTutorial>。方便大家下载查看，可以自行修改，唯一的要求是开源。

另外教程也可移步[我的博客](#)，扫面下方二维码关注我的微信公众号亦可



本文档使用的模板来自于 ElegantBook，网址已经停止维护了，但是其中一个作者的博客还在更新[Ethlisan](#)，感谢作者提供模板

# 第 1 章 使用 Capture 画原理图



## 第 2 章 建议规范



以下规范是在画板过程中总结出来的，分为三类：必须遵守、强烈建议和个人建议。请尽量按照这个规范来画，如有不当请指正。

### 2.1 Allegro 配色方案

Allegro 提供不同的颜色将不同的 subclass 的颜色区分开有助于 PCB 设计，不同人有不同的配色标准，本无不可，但是相互之间查看电路板还是会带来障碍，因此在这里提供一种配色方案，希望大家都采用。

#### 2.1.1 ETCH

Top 红色

Signal1 天蓝色

Signal2 橘黄色

Signal3 粉红色

更多的内电层颜色请自定义鲜艳的颜色

Bottom 层深蓝色

#### 2.1.2 Net

目前只规定 GND 的颜色为青青草原色

电源请自行选择鲜艳颜色，可以和 ETCH 的颜色重复

#### 2.1.3 其他辅助层

Top 层丝印白色, Bottom 层丝印黄色

## 第 3 章 使用 Allegro 画 PCB



## 第 4 章 平时的一些经验

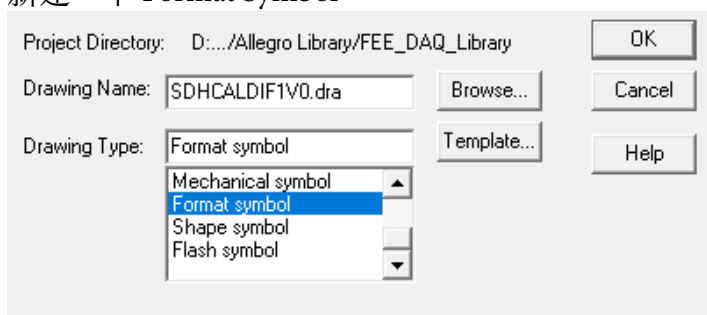


### 4.1 制作 LOGO

1. 需要准备一张 bmp 格式的图片，如下图所示

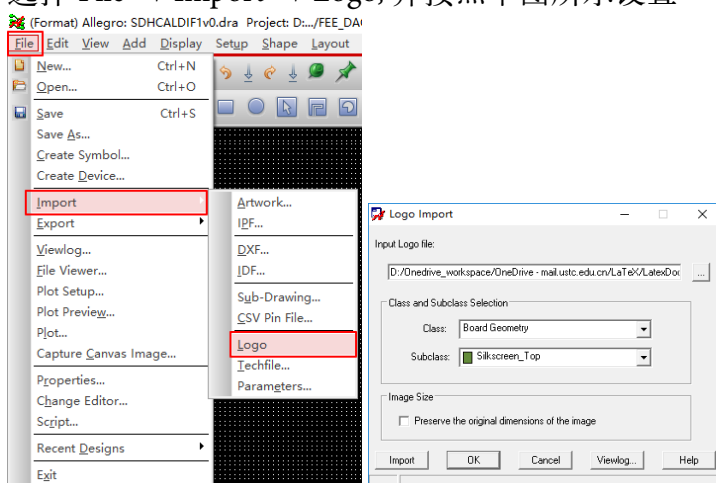
SDHCAL FEB V1.0

2. 新建一个 Format Symbol

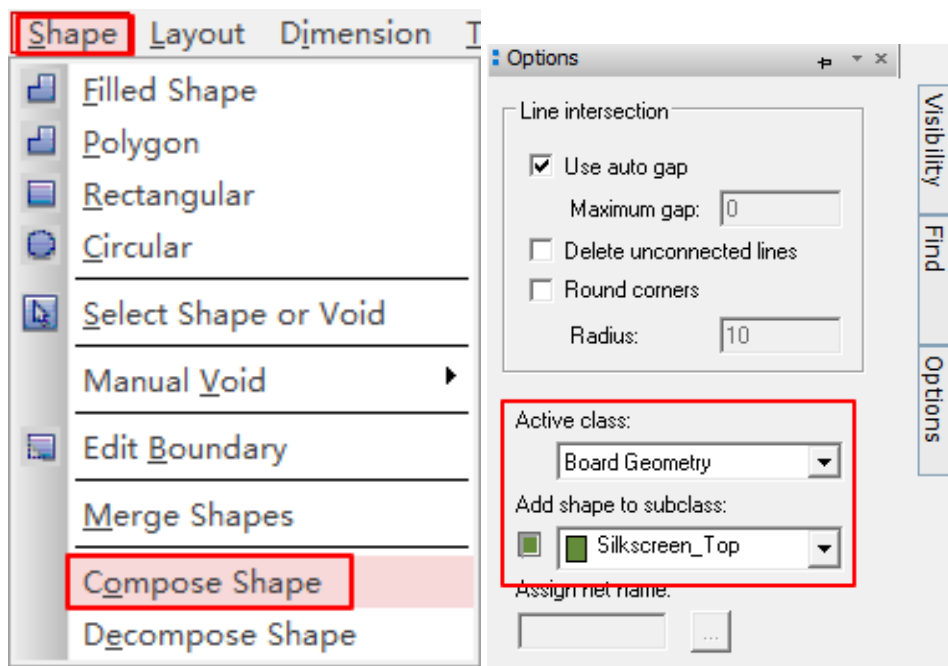


3. 设置好画布和栅格点大小，按照经验最后 LOGO 的大小是画布大小的一半，按照自己需要的大小设置栅格点即可

4. 选择 File → Import → Logo, 并按照下图所示设置



5. 使用命令 Shape → Compose Shape 并在 Option 中设置 Class 和 Subclass 分别为 Board Geometry 和 Silkscreen\_Top



6. 保存。大功告成了

## 4.2 Skill 教程

### 4.2.1 Allegro skill 介绍

- Skill 是 Cadence 提供的可二次开发的语言。语法同 C 语言类似。在设计中使用 skill 可以大大简化 PCB 绘制流程，还可以定制各种各样的功能
- 本文只对 skill 使用做一个简单的介绍，进阶的方面以后学会再做笔记
- 推荐一个网站：[Allegro Skill](#)，使用的 skill 和介绍均来自此网站

### 4.2.2 Allegro skill 设置方法

以一个 skill 为例(ch\_via\_net)，这个 skill 的功能是将电路板中的过孔的网络修改为任意一个网络。

1. 先从任何一个地方获取到这个 skill 文件 ch\_via\_net.il
2. 将文件放置在一个文件夹中，不含中文和空格。如我放置在 D:\Cadence\skill 中，方便日后管理
3. 在环境变量文件夹中找到 allegro.ilinit 文件，这个文件一般在 C:\Users\\AppData\Roaming\SPB\_Data\pcbenv 中，表示计算机用户名。





4. 编辑这个文件，在文件中加入如下的代码

```
setSkillPath ( buildString ( append1 ( getSkillPath () "D:/
Cadence/ skill ") ) ) ; 设置 skill 所在路径
load ( "ch_via_net.il" "www.allegro-skill.com" )
```

; 载入 skill 前一个参数是 skill 文件，后一个是密码。

5. Skill 设计结束，可以在工程中使用。
6. 最好为 skill 的操作设置一个快捷键，不然使用中不会很方便。
7. 可以自定义 allegro 菜单，将自己添加的 skill 加入菜单中方便使用在安装路径 D:\Cadence\SPB\_16.6\share  
textbackslash pcb\text\cuimenu 中找到 allegro.men 文件，该文件为加载目录文件。  
在目录的最后一个 end 前加入如下代码：

```
POPUP "My_Skill"
BEGIN
MENUITEM "&Chang Via's Net", "ch_via_net"
END
```

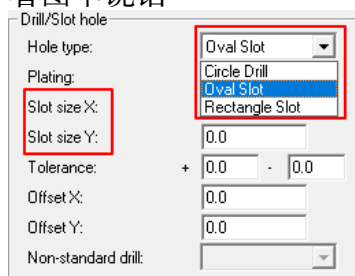
### 4.2.3 AlignTool

安装方法：

1. 下载 AlignTool1.0.zip 并解压到电脑中。
2. 在解压出来的文件夹中直接点击 install.bat 进行安装，不需要手动进行安装。
3. 重启 allegro，输入命令 aln 运行。

## 4.3 制作异形过孔

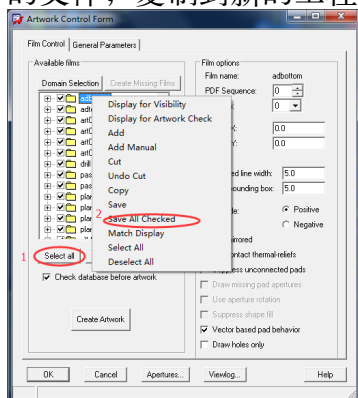
看图不说话



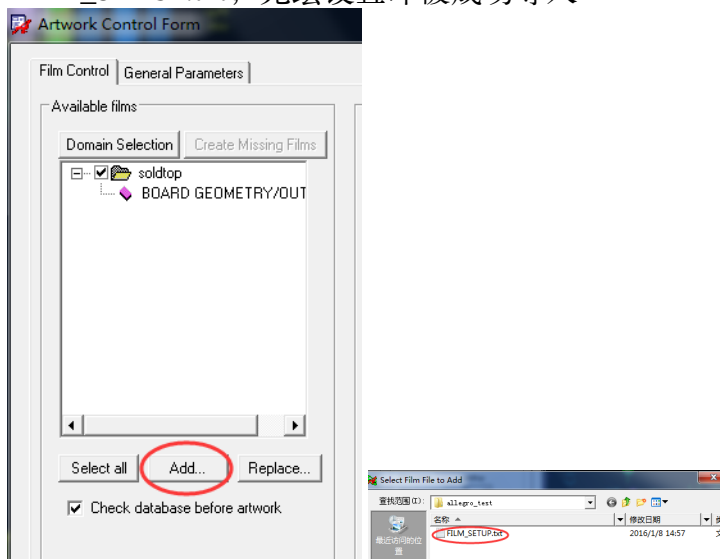
## 4.4 从之前的工程中导入光绘设置

在画电路板完成后需要生成光绘文件，如果每次都手动添加光绘设置会显得非常细碎，同时可能会有失误，如果此时正好有一个相同层数且布局相同的.brd 文件，就可以从中导出以前的光绘设置，大大减少工作量。下面就是操作步骤

1. 先打开已经生成过光绘的.brd 工程的生成光绘页面
2. 点击 Select all 选中所有的光绘层，然后对着其中一个层右键单击
3. 在出现对话框中选择 Save All Checked
4. Allegro 会在工程文件夹下面生成一个 FILM\_SETUP.txt 的文件，这便是我们需要的文件，复制到新的工程目录下



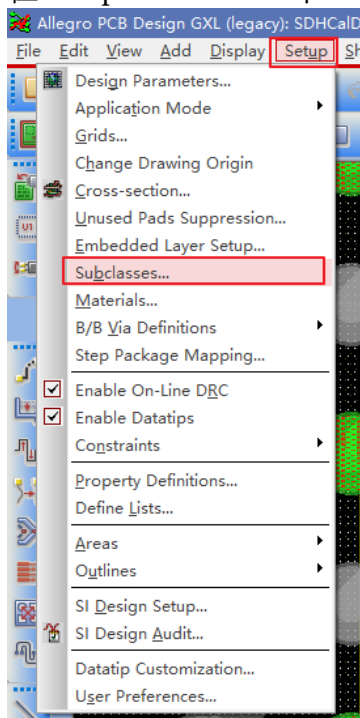
5. 在新的工程中生成光绘页面点击 Add，在弹出对话框中选择刚刚复制过来的 FILM\_SETUP.txt，光绘设置即被成功导入



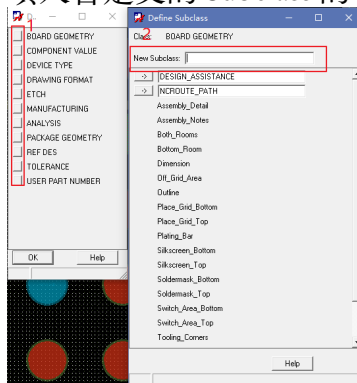
## 4.5 新建 SubClass

Allegro 自己默认了许多的 Class 和 Subclass，这些 Subclass 都是在画电路板的时候必须的。有的时候画电路板需要一些辅助线，比如说分割 FPGA 的 bank，摆放器件的时候也需要一些辅助线。以前的做法是放在丝印层，但是这种方法并不是设计安全的，如果忘记删除，会留下一些不好看的痕迹，最好的做法是创建一个自己的 Subclass 来摆放它们，下面是如何自定义 Subclass

### 1. 在 Setup→Subclasses 中



### 2. 然后在弹出的窗口中选择一个 Class 点击，然后在弹出的框中的 New Subclass 中填入自定义的 Subclass 的名称

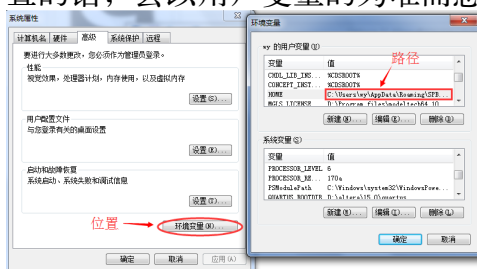


## 4.6 Allegro 设置快捷键

Allegro 中本身默认了一些快捷键，但是使用起来不太方便，大多需要两个键一起组合，本教程介绍如何修改快捷键

### 4.6.1 快捷键介绍

先说一下 Allegro 的变量文件，一共有 2 个，一个是用户变量，一个是全局变量。用户变量文件的位置，通过系统环境变量设置：系统属性-高级-环境变量，其中的 Home 值就是 env 所在目录。要注意的是，这里也有两个变量，一个是用户变量一个是系统变量，在用户变量里设置了 Home 之后就不需要在系统变量里再设置了，如果同时设置的话，会以用户变量的为准而忽略系统变量。



### 4.6.2 修改快捷键

1. 在 C:\Users\\* \AppData\Roaming\SPB\_Data\pcbenv 下面，编辑 env 文件，用任意文档编辑器打开即可,\* 是用户名
2. 修改快捷键有两个命令一个是 alias, 另一个是 funckey,
3. alias 可以用来指定除字母以外的其他按键，举例如下

下面是我的一些快捷键



```

1 source $TELENV
2
3 funckey d done
4 funckey v delete
5 funckey n next
6 funckey g move
7 funckey q add connect
8 funckey w slide
9 funckey c cancel
10 funckey r rotate
11 funckey m mirror
12 funckey Esc oops
13 funckey l Add Line
14 funckey e copy
15 funckey t Text Edit
16 funckey y Add Text
17 funckey f Create Fanout
18
19 funckey 11 'pop swap;subclass top'
20 funckey 22 'pop swap;subclass top;+'
21 funckey 33 'pop swap;subclass top;+;+'
22 funckey 44 'pop swap;subclass top;+;+;+'
23 funckey 55 'pop swap;subclass top;+;+;+;+'
24 funckey 66 'pop swap;subclass top;+;+;+;+;+'
25 funckey 77 'pop swap;subclass top;+;+;+;+;+;+'
26 funckey 88 'pop swap;subclass top;+;+;+;+;+;+;+'
27 funckey 99 'pop swap;subclass top;+;+;+;+;+;+;+;+'
28 funckey 00 'pop swap;subclass bottom'
29
30 funckey a1 'pop bddrill;pop swap;subclass top'
31 funckey a2 'pop bddrill;pop swap;subclass top;+'
32 funckey a3 'pop bddrill;pop swap;subclass top;+;+'
33 funckey a4 'pop bddrill;pop swap;subclass top;+;+;+'
34 funckey a5 'pop bddrill;pop swap;subclass top;+;+;+;+'
35 funckey a6 'pop bddrill;pop swap;subclass top;+;+;+;+;+'
36 funckey a7 'pop bddrill;pop swap;subclass top;+;+;+;+;+;+'
37 funckey a8 'pop bddrill;pop swap;subclass top;+;+;+;+;+;+;+'
38 funckey a9 'pop bddrill;pop swap;subclass top;+;+;+;+;+;+;+;+'
39 funckey a0 'pop bddrill;pop swap;subclass bottom'
40
41 alais mQ add line
42 ### User Preferences section
43 ### This section is computer generated.
44 ### Please do not modify to the end of the file.
45 ### Place your hand edits above this section.
46 ###
NORMAL C > \Users\wyu98\AppData\Roaming\SPB_Data\pcbenv\env
Already at oldest change

```

## 4.7 钽电容画法

此法又叫手动焊盘增大术，名字来源于[Haolei Chen](#)

### 4.7.1 电源去耦电容的作用

通俗来说滤波电容的作用就是保证芯片供电量增大时供电能够保证，一般使用一个小的陶瓷电容配上一个大的钽电容。打个比方，两个电容就像水窖和水库，当芯片耗电量突然增加时，首先从陶瓷电容上放出电荷，WTW.

### 4.7.2 钽电容画法

钽电容的焊盘一般都比较大，如果还是用 6mil 或 8mil 的走线将其连到相应的电源和 GND 上势必会造成较大的走线电感，和钽电容的寄生电感一叠加就雪上加霜了。更重要的是，用细的走线势必要用小过孔，小过孔不仅增加电感，过大电流能力还不好，那就用大过孔 + 粗走线，如下图所示

一个钽电容用 40mil 的走线连到一个 40mil 的过孔上，问题是解决了，但是大的地孔会将地平面和电源平面打出一些洞，如果不幸有走线在这些洞附近，其地回路必然受到影响。还有更优的选项，在焊盘上铺上一层铜，然后在铜皮上打许多小过孔，这样寄生电感更小还可以塞孔，如下图所示，相当于把钽电容的焊盘增大了一部分用于打过孔。这样的方法同样适用于其他需要和内电层良好接触的表贴焊盘。



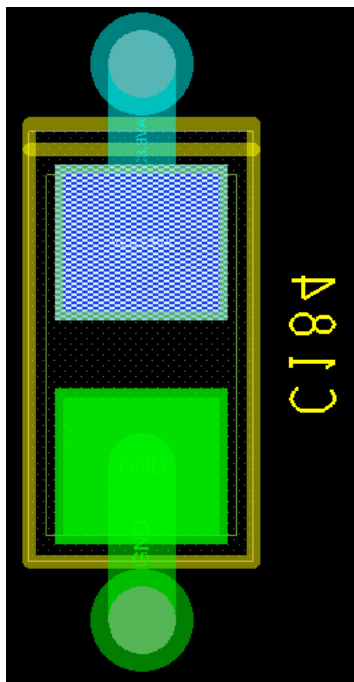


图 4.1: 用 40mil 走线和 40mil 过孔连接钽电容

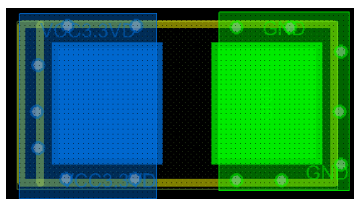


图 4.2: 在钽电容焊盘上铺铜然后打小孔

## 4.8 盲埋孔设计

