硬件工程师学习：

-------

了解什么是直流电，什么是交流电，先从直流电开始～

1:先学习电阻（很枯燥）

2:学习电容，（有点意思了）

3:接入二极管，（诶？好神奇）

4:简单的入手低压，低频率交流电。

5:交流到直流的转换

6:开始进入三极管（终于开始3只脚的家伙了）

7:学习电子开关，一种是硬开启，硬关断；另一种软开启，软关断。

8:着手准备运放，①学习比较器，②学习放大。

9:回头，开始加入电感（又回到两只脚了）

10:简单的功能IC（开始翻PDF了）

11:总算都熟练了，琢磨怎么把这一大堆玩意儿瘦身呢[思考]

12:引入单片机概念

13:门电路走起

14:流程图、逻辑图走起

15:数字、模拟之间的转换走起

16:了解编程语言，各个定义的意思以及如何使用

17:开始膨胀～（先回点本再说）?

------

其实本人硬件学习之路和up里讲的基本上一样，先去粗略学一下基本元器件，然后去学51单片机，再看懂原理图，学习其中功能电路，最后学习焊接和eda软件使用。

当初学eda软件时不会和你说元器件选型，只要照着画就好了，画好拿去打板（反正jlc白嫖），看到自己一步一步画出来的电路板真正拿在手上的感觉，心里一番滋味，然后自己研究别人的电路，抄别人的电路模仿做出来，就慢慢学会硬件有问题的时候该如何调试和示波器万用表频谱仪使用。看着别人的电路有什么功能，怎么实现的，然后自己再去马云网上看有没有现成模块卖，买回来研究，慢慢的就知道电路芯片如何选型了，然后也从店家那送的数据手册慢慢看懂芯片数据手册。（其实做单片机的时候就应该知道的，当时也只看懂了流水灯的138译码器）再然后接触各种芯片制造商，选型的时候也不会那么单一，比如以前ldo总是喜欢用ams1117，其实也有很多国产优秀的电源芯片制造商。

----------------------------------

如何学习硬件设计——实践篇

实践篇的教程，操作类的东西，很难用文字来表达。那只能把最关键的口诀写下来，让大家在实践中体会了。

一般缺乏经验的工程师或者学生，拿着一个项目任务书，或者一个成品的电路板的时候，往往会感觉到，根本无从下手。主要原因是，知识储备不足，少实践少动手。

但也不用着急，这是需要慢慢积累的。同样，不用担心东西太多，不知道学到什么时候才能独当一面，因为很多东西都是相通的。

下面介绍硬件设计的实践路线。

一、初级实践篇。

1、焊接。

首先看一下杜洋前辈的焊接视频。我当年也是看了他的视频才学会焊接的。

关键的地方视频中也会提到，这里大概地说一下（洞洞板的焊接就不说了）。

拖焊的时候，先对齐芯片，再上锡固定一个角，然后在另一侧加满锡，最后整个芯片都加满锡。把板子拿起来，倾斜30度左右，再用烙铁加热，把变成液体的锡吸起来，甩掉，直到把所有锡都吸走为止。烙铁的温度要调好，我一般用350摄氏度。重点要体会，锡变成液体的时候，会像水一样受重力作用向下流，还有，烙铁头表面是有吸力的，所以在整个焊接过程中，都不要用力刮锡的。如果焊的时候，操作起来不顺手，可以转动板子。

关于BGA的焊接，一般是不建议手工操作的，因为成功率不高，推荐用返修台。这里说一下BGA手工植球的操作流程。

先用万能植锡钢网（这是最落后的工具，除此之外还有植锡台，不过挺贵的），跟BGA对齐，再用胶布把BGA和钢网粘住固定好。先加锡膏，再用风枪吹一会（风枪的风速和温度可以调低一点），锡变亮的时候，再用手术刀，把多余的锡刮走。如果锡球不均匀的话，再重复上一步，直接锡球均匀为止。撕掉胶布，用手术刀把BGA撬起来。

2、仪器仪表的使用。

a、万用表。为什么起这个名字？因为对于高手来说，万用表是几乎是万能的。一般也是用它来测电压、电流和电阻。

b、示波器。现在都用数字示波器，一个auto键，可以轻松搞定，而且还带FFT的功能，可以使用频域分析法，是硬件工程师必须掌握的神兵利器。示波器还有个小众的功能，就是李沙育图（测相位差和测频率用的）。此外，还要学会用示波器测开关电源纹波。

c、数字电桥，也叫LCR、LCZ测试仪。用它可以测电感值、电容值、电阻值、Q值、D值等，精度比一般的万用表要高。

d、信号发生器，也叫函数信号发生器。可以输出正弦波、方波、三角波、已调信号。用法比较简单，但是射频信号发生器，就要注意了，在输出信号之前，一定要做好阻抗匹配，不然信号反射的话，有可能会损坏信号发生器。

f、频率计。用法比较简单，不再多说了，有的信号发生器还增加了频率计的功能。

g、矢量网络分析仪，也叫网分仪。用于测量射频电路的S参数矩阵，还可以显示史密斯圆图。每次使用之前都要校正一下频率点。

h、频谱仪。也就看一下频谱，也有示波器的功能。

还有一些小众仪器就不说了，像漏电流测试仪、电表等。

3、维修。

首先肉眼观察一下板子，看有没有虚焊、短路或者缺少元件。有就修，没有就下一步。

然后用万用表测一下各组电源，看有没有短路。有就修，没有就下一步。

给板子上电，看各组电源电压是否正常。有就修，没有就下一步。

到了这一步，你必须对板子的整体设计有一定的认识，或者你得背下前辈们的经验（背经验的往往觉得硬件很神秘，这是我不推荐的做法），不然没法修好。先对板子的各个功能分好模块，从现象判断哪个模块出问题，断开可疑的模块，来排除可疑点（像侦探一样）。有一块好板的话，就很好办，直接对照着测各元件的电压（或者对地的电阻值）就能解决了。用万用表只能解决一些简单的问题，要想彻底修好，手上一定要有示波器，因为像晶振受到干扰之类的，用万用表是测不出来的。

4、调试。

调试，一般是自主设计的电路，没经过验证，需要自己去验证，这是非常需要扎实的理论基础。调试也是硬件工程师最容易累积经验、含金量最高的技能之一。如果前期遇到棘手的问题，可以暂时先放下，等后来水平再高一层，就会解决的了，所以千万不要钻牛角尖，这只会浪费更多的时间。调试的技巧需要长时间的积累，放在前面，是让大家有所重视。

调试方法，多种多样，视情况而定，不能一概而论，笔者总结了以下几个方法：

a、示波器测量。当然，首先你得清楚你设计出来的电路，会出什么样的波形，才知道测出来对不对，也就是说，理论不行的，根本无法调试。

b、对照验证过的电路。如果手上有一块好板，而需要调试的电路里面刚好有好板的电路，可以拿好板来飞几根线验证一下，排除可疑点，这里跟维修的方法一样。

c、仿真。其实在设计电路的时候，能仿真就先仿真了，如果实物做出来，还是有问题，也可以仿真一下。如运放电路的参数、不确定的电阻串并联等等。

d、镊子短路。在你怀疑时钟是不是干扰到其它信号的时候，可以用镊子把时钟引脚跟地短路（只要是弱信号，跟地短接一会都不会烧板子的，放心），以排除可疑点。还有复位的问题，也可以用这个方法。

e、信号发生。比如一个运放电路，输入和输出均受干扰了，那么你就可以用信号发生器或者开发板，来输出一路干净的信号，这样可以排除可疑点。

f、软件调试。如果板子上，有CPU就可以用串口调试，有FPGA就可以用嵌入式逻辑分析仪，这样可以确定是芯片内部还是外部的问题。

g、观察现象。信号都在板子上跑了，直接观察是观察不出来的，这个时候，可以引出信号线，接在可观察的设备上。如：调试音频放大器的时候，就可以接一路信号，到一个现成完好的功放上面，通过听声音来观察现象。当然，你可不要只想到功放，还有其它可观察的设备或者元件，像LED灯、显示器，甚至是收音机，只要能派得上用场的都可以。

二、中级实践篇。

1、仿真软件的使用。

常用的仿真软件也就那几个，proteus、multisim、labview、pspice、ADS、saber等，其中大多数是用spice仿真模型。

a、proteus。这个软件很适合仿真单片机，元件库也挺多的，但是有个致命的缺点，就是太智能了。单片机不接电源、不接晶振也能正常工作，这跟实际有很大出入，所以笔者建议学单片机，还是用开发板吧。

b、multisim。这个软件很适合仿真模拟电路，其实它本质是spice仿真，只是界面做得简单很多，适合初学者使用。虽然有8051的库，但是，不适合仿真单片机，仿真起来很慢。元件库其实并不多，像0805的三极管，它都没有，这时候只能用其它的三极管（2N2222等）代替一下，要不，就自己做这个元件库。multisim还可以跟ultiboard配合使用，实际板级仿真（连同PCB，一起仿真）。

c、labview。这个软件功能非常强大，可以仿真模拟、数字电路、也可以做上位机（如：虚拟仪器等）。最具特色的，就是图形化输入，鼠标施几个东西就 可以仿真了。

d、pspice。这个软件是cadence或者叫SPB开发套件中的一个软件，一般是在capture中调出来的。使用capture就可以不用输入spice的点命令，非常方便。其中，pspice的图表要比multisim的要好看一些，比如，测几个节点的电压，在pspice一张图就看得很清晰了。

e、ADS。这个ADS是指Agilent的Advanced.Design.System，而不是指ARM编译器ADS1.2。ADS可是电路仿真的神器啊，功能非常强大，一般是仿真高频、射频、微波电路用的，当然，集总参数电路也照样可以仿真，但是不太适合初学者。

f、saber。这个软件是专门仿真电源电路用的，笔者暂时没用过，不做评价。

2、电路设计软件的使用。

主流的电路设计软件有三个：altium designer、PADS、Cadence，当然还有些小众的，像eagle。这里只介绍主流的三款软件。

altium designer（简称AD），以前的版本是protel 99se，protel DXP，用法都大同小异，很适合初学者使用，3D渲染效果最好，同时也是学校里教得最多的软件。但是，很多公司反而不用这软件，因为用它画多层板的话，电脑会很卡，而且公司里面用的人多的话，可能会收到altium的律师函。可以用它来做FPGA开发，并进行板级仿真。适用于小规模的PCB。

PADS，以前的版本是power PCB，分成三个组件：logic（原理图）、layout（布局和设置规则）、route（布线），最具特色的功能是：使用极坐标放置元件和自动布线（这个自动布线可没有AD那么烂）。适用于中小规模的PCB，但是logic相当不好用，所以有些人用orcad+PADS来弥补这个缺点。适用于中小规模的PCB。

Cadence（也叫SPB）是个系统级的套件，除了画原理图、PCB之外，还可以画版图、仿真电路、仿真SI/PI等。Cadence公司收购了orcad，目前画原理图的是用capture（也叫orcad），画PCB是用allegro，仿真电路的是pspice（从capture里面调出来的），仿真SI/PI的是Sigrity（需要另外安装）。用capture画原理图是非常爽的，比如，画个芯片的原理图库，你可以用excel写好（引脚号和部分引脚名，像D0~D7，鼠标拖一下就出来了），然后copy到capture里面，再做少量的调整就可以了。但是用allegro画封装就比较烦琐，需要事先画好焊盘，才可以画封装。适用于中大规模的PCB。

3、其它软件的使用。

画板框用的autoCAD、画3D封装的solidworks或者pro-e、科学计算的MATLAB。

autoCAD的基本用法还是比较简单的，在有人教的情况下，半小时可以入门，对于硬件工程师来说就画一下板框，保存为DXF格式，再导入到PCB设计软件。同时，DXF也是硬件工程师与结构工程师交互的文件格式。

相对于pro-e来说，solidworks更加易学易用。用这两个软件都可以画元器件的3D封装，再把PCB导出为stp格式放到solidworks当中，这样，还没打板就可以看到整机的效果图了。学3D软件还有个好处，让你更清楚板子安装的情况，像定位孔、插座、接线等，这样设计出来的PCB不容易因为结构问题而无法安装，这是很多硬件工程师容易忽略的地方。

MATLAB，任何的计算，都可以用它。简单的计算，像电阻分压、滤波器的截止频率等，复杂一点，像定向耦合器的参数计算、复杂运放电路的建模等，用MATLAB都可以轻松解决。这里还推荐一个网页版的计算工具。

三、进阶中级实践篇。

1、基本电路单元的计算、仿真与验证。

诚然，不管一块电路板有多复杂，都可以按照功能来划分为若干个模块，而这些模块还可以再划分为众多的电路单元。所以，首先要掌握最基本的电路单元的设计。这些电路单元，都可以在数电、模电、电力电子技术、高频电子线路、单片机、电子测量技术当中学到，先搞懂教课书上经典电路的计算、仿真与验证。不要以为书上的公式简单，但是实际操作起来，又是另一回事。比如，书上的反相放大电路，是双电源的，用单电源就要加偏置，还得考虑带宽增益积、摆率等。这里主张先计算，再仿真，后实物的操作流程，同时，这也是一个需要长期累积的过程。

2、掌握单片机。可以参考本博客中的《如何学习单片机》。

3、芯片的使用与互连。

在理论篇里面没有写到电子专业英语，在这里就要用到专业英语了，你可以看英语教材，也可以用翻译软件。这里必须提到的一点是：英语不好导致无法阅读datasheet的，都无法做电路设计。因为你总得会用到一块陌生的芯片，总会遇到没中文资料的情况。基本上能看懂datasheet的，都能把芯片用起来，其实也是抄datasheet上面的参考电路的，剩下的，就是芯片互连。

芯片互连，就是接口技术，也是单片机里面会讲到的。5V的ADC跟3.3V的单片机互连，这就要看电平、和信号的传输速率了。3.3V单片机跟12V开启电压的MOS管互连，加个三极管，做电平转换就可以了。两块3.3V单片机IO口推挽输出互连，串个100R电阻，防止代码操作不当而烧坏IO口。

此外，还要掌握常用的总线协议。比如RS232、RS485、SPI、IIC、CAN、LIN、zmodem、USB、PCIE、TCP/IP等。

四、高级实践篇。

在这里，相信你已经把一些基本电路，熟捻于心，也会分析一些简单的电路。但是，你总会遇到一些奇葩的现象。没错，你是时候要考虑SI、PI、EMC、EMI了。不要被这些貌似很高端的名词吓倒，分析起来，也是前面学到的电路原理，只是考虑问题的角度不同罢了。

1、SI，信号完整性。这部分的内容对PCB的布局、布线影响较大。

a、使用阻抗匹配减弱过冲、下冲、振铃的影响（某些射频电路也对阻抗有要求，如：天线等）。

b、差分线应该尽量靠近以减少差模干扰。

c、去耦电容要尽量靠近芯片的电源管脚。

d、继电器等大功率器件应该远离晶振等易被干扰的元件。

e、对重要的信号线，包地。

f、尽量远离时钟线（时钟也可能成为干扰源）。

g、信号线的返回路径应该尽量短。

信号完整性要注意的地方，还是挺多的，具体可以参考王剑宇的《高速电路设计实践》。

2、PI，电源完整性。要保证电源的完整，就是防止电源电压的波动，具体可以参考本博客的《去耦电容的作用》。

3、EMC/EMI，电磁兼容性和电磁干扰。这两个名词看起来有点高大上，其实就是不干扰别人和防止被别人干扰的问题。EMC/EMI的问题可以归结为SI的问题，但是EMC有一套验证的标准，所以还是起了不同的名字。

推荐《Cadence高速电路设计：Allegro Sigrity SI/PI/EMC设计指南》。

五、总结。

1、千万不要以为把某些口诀当秘笈地记下来，就以为练成了神功，这都是不现实的。前期的学习都必须以理论为核心，少量的实践以帮助理解理论，后面就可以逐渐增加实践，理论和实践是相辅相成，缺一不可的。

2、当硬件电路出了问题，工程师每一步的操作，都是以理论作为指导思想的。

3、千万不要害怕出错而不敢做板。硬件工程师都是不断地犯错、改正、总结，才慢慢地成熟起来，减少犯错的概率。不知道错的话，也意味着不能积累经验。

4、本文没有提及生产、测试方面的问题，如：线材、PCBA、BOM、拼板、测试夹具、打磨芯片、包装（QC标签、易碎纸、说明书）等。

5、因为大多数的电路功能都依靠于芯片来实现，画原理图几乎都是抄datasheet的，所以硬件工程师最具含金量的技能是PCB和调试能力。

6、因为硬件工程师也常常需要和软件工程师交流，所以，为了方便交流，你还得学习ARM、FPGA、DSP等相关知识，只是侧重点有所不同而已，不然会给工作上带来一定的麻烦。