

1. C++开发训练

1.1. 基础背景

本科就读于广州大学，专业是计算机科学与技术。本科期间，主要使用的语言是 C++，做过一些课程设计，使用 MFC 做过一些简易的可视化界面。考研期间，主要学习了数据结构、计算机组成原理、操作系统、计算机网络。但对于 C++ 的使用，多数也只是敲算法题，只用到 C 语言加 STL 库，很少会用到类和对象这些 C++ 的特性去写代码，更没有接触过大型的项目开发。

1.2. 学习过程

C++ 的学习路径主要按照水滴计划的安排来走。第一周学习了各种各样的插件，比如 Clang、Google Test、Doxygen、CMake，不过对于这些插件的理解也仅仅停留在字面意思，没有实际使用过。

第二周开始看书，主要阅读《C++ Primer Plus》第 7-13 和第 16 章，这本书里面很多代码，我选择静下心来慢慢啃，比较耗时间。但结合理论知识，看完里面的代码之后，还是有所收获的。看完书后我完成了每一章的课后习题，PR 到仓库了。紧接着我看了《现代 C++ 教程》，这本书比较精炼，总结了一些现代 C++ 的新特性，值得一看。

第三周开始做 A* 算法大作业，先实现了整个算法的主体流程，然后再将这个算法部署到现代 C++ 模板中，最后再编写 Google Test 测试用例，使用 Doxygen 插件添加注释。

1.3. 学习成果

学习笔记链接已经 PR 到仓库里了，可以在 iTraining 仓库里查看。按照水滴计划的路线学完 C++ 之后，收获还是比较大的。以前写算法题，基本就是一个 cpp 文件就可以解决问题，也懒得分别写 .h 文件和 .cpp 文件。直到做 A* 大作业的时候，我尝试把问题拆解成三个类，再写 CMakeList.txt 将多个 .h 文件和 .cpp 文件统一编译。做完这个作业我才体会到什么叫 C++ 的项目，也才体会到将类封装好的方便性。

同时，学习期间，我对 VSCode、Linux、Git、SSH 的使用更加熟练了，以前都是用 Visual Studio 来写代码，没接触过 VSCode，没远程使用过服务器，现在用了才发现真香。

1.4. 总结和建议

刚开始配置环境和学习各种插件的时候，可能会觉得云里雾里，其实不用太担心，可以先尝试往后学，到了后面做大作业的时候，尝试去使用它们，很快就能学会了。看书的话，我建议要结合代码来看，不要把大段大段的代码跳过不看，不然印象会不够深刻。我做课后习题的时候，也会时常回去翻看书上的代码，看看人家是怎么写的，我再模仿着他的代码风格去写。

2. EDA 学习和实践

2.1. 基础背景

本科期间学过数字电路，还有一点记忆，但是没有接触过 EDA 的相关知识，也没学过数字集成电路，完全是小白。

2.2. 学习过程

学习 EDA 的前两周基本都在看书，把《超大规模集成电路物理设计：从图分割到时序收敛》看完了，《数字集成电路物理设计》看了一部分。个人感觉，前者有比较多布局和布线相关的算法，每一章前面还会先详细介绍一些相关的概念和名词，对新手比较友好，而后者更偏向于工艺设计，我一开始看得似懂非懂，后面就没继续看下去了。

看完书之后，开始 EDA 的实践部分。我阅读 iEDA 的官方手册，跑通了 iEDA flow，通过这个过程大概了解了从网表到版图的全过程，熟悉了每一个阶段生成的格式文件。之后开始做布局合法化大作业，先是阅读了 Abacus 算法和 Tetris 算法的论文，搞清楚了它们的原理和二者的差别。iEDA 里已经实现了 Abacus 算法，只需将里面的几个函数改动一下，就可以实现 Tetris 算法。用这两个算法分别去跑，可以看出二者在速度上还是有很大区别的。

2.3. 学习成果

学习笔记链接已经 PR 到仓库了，可以在 iTraining 仓库里查看。这三周主要学习芯片的后端设计流程，对逻辑综合、布图、布局、时钟树综合、时序收敛有了大致的了解，通过《图分割到时许收敛》这本书，还学习了一些关于网表系统划分、芯片划分、芯片规划、全局布局、详细布局、全局布线、详细布线的算法。

在 EDA 实践过程中，我学会了如何跑通整个 iEDA flow，以及使用点工具，单独跑某个步骤，跑完后在 report 中查看结果，记录参数。改 Abacus 算法的时候，也深入地去理解了

算法背后的数据结构，以及每个函数负责做什么。经过 EDA 实践，我对书上的知识有了更加深刻的理解。

2.4. 总结和建议

最开始看书的时候，不必太过于关注细节，要把握好每一部分知识学习的深度，否则会导致进度慢，浪费时间。对 EDA 的学习要分层次，首先要把握好整个芯片设计流程，知道后端设计流程大概分为哪几个主要步骤，每个步骤需要做什么工作，做完这些工作之后会得到什么。对于整体的把握，我认为 iEDA 的官网上关于 EDA 知识的讲解就比较好，讲得比较精炼易懂，先看完这些知识再去看书效果会更好。

自我感觉学习 EDA 的过程中，学习重心出现了问题，我花了较多时间去看书，但感觉收获甚微，学不进脑子，忘得比较快。我认为应该要把更多的时间花费在实践上，认真学习 iEDA flow 整个流程，去看看每个流程做什么工作，需要输入什么格式的文件，输出什么样的文件，每一个点工具后面的数据结构和算法是怎么样。可能一开始我这样学，先把知识体系建立好，效果会更好。

3. 选学部分（AI）

3.1. 基础背景

本科上过人工智能导论、机器学习、计算机视觉这些课，接触过一点点 AI 相关知识，毕设做的是目标检测类的算法，但懂得并不多，还没系统学过深度学习。考研期间学过高等数学、线性代数、概率论，一些基本的数学知识还有一点印象。本科期间用 python 比较少，numpy、panda 这些库不太熟悉。

3.2. 学习过程

目前在 b 站上看李沐老师的“动手学深度学习”，打算系统地学一下深度学习。

3.3. 学习成果

3.4. 总结和建议