

# “一生一芯”太理工作室一组寒假研学第二次学习路线

学习情况：经过上周理论学习为主，本周我们要重视实践部分了

学习目标：Verilog实践练习和LCTHW数据结构实践

## 学习任务

### 1. 数字设计

我们在这之前学习了C语言，makefile，markdown，或许有人还学了一部分shell脚本语言；但我们今天要开始学习一门全新与众不同的语言——RTL硬件设计语言Verilog

什么是Verilog？

Verilog HDL (Hardware Description Language) 是一种硬件描述语言，用于从算法级、门级到开关级的多种抽象设计层次的数字系统建模。现实生活中多用于专用集成电路 (Application Specific Integrated Circuit, ASIC) 和现场可编程门阵列 (Field Programmable Gate Array, FPGA) 的实现。

#### ? 看不懂定义？

其实很简单

##### 1.为什么与众不同？

因为这就是**硬件编程**，我们需要更多地关注于电子元件和电路的设计，**实现硬件设备的控制及其与软件的交互**。相比之下，软件编程更多地涉及编写代码来构建应用程序，这些程序被设计为运行在硬件上。因此我们编程时关注的点完全不一样

##### 2.为什么有这样一门语言？

还记得当时的宣讲会吗？随着**集成电路规模越来越大，电路越来越复杂**，传统的画图或连线的设计方法已不再适用；此时就诞生了这样一门硬件编程语言，利用**代码生成庞杂的电路**

##### 3.他能做什么？

a.设计者可以通过编程来实现**加法器、计数器、寄存器等基础电路**，同时也能够设计出**CPU、存储器等复杂系统**。这个过程中，设计者不仅能够提高自己的逻辑思维能力，还能学会如何优化电路设计，使之高效运行。

b.也可以去编写**仿真测试**。验证自己设计的电路没有致命问题，如果开始生产才出现问题，那就浪费了极大的金钱成本。

c.支持**FPGA**(Field-Programmable Gate Array)和**ASIC**(Application-Specific Integrated Circuit)开发上。这两种技术都是现代电子设备中非常关键的部分，它们使得定制化的高性能电子系统设计成为可能。

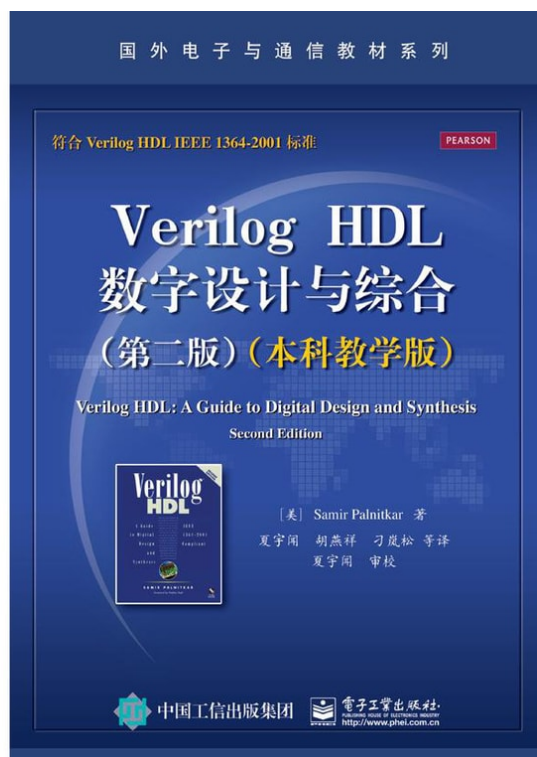
d.多领域发展，包括但不限于**嵌入式系统、人工智能、物联网**等前沿技术领域。**嵌入式系统**中经常会用到FPGA来处理特定的高速或并行计算任务，而Verilog则是实现这一目的的关键工具。在**人工智能**领域，Verilog可以用来设计高效的机器学习硬件加速器。**物联网设备**也常常需要低功耗且高性能的定制电路，Verilog编程在这里同样发挥着重要作用。

你现在应该明白这一门语言的特殊性和重要性了^\_^，如果还不明白的话，可以看一下下面这个视频：  
30分钟了解verilog特性：[verilog入门视频](#)

## 内容

**理论**方面学习资料推荐：

Verilog HDL数字设计与综合（**群文件中有电子版**）




本周学习章节：**第七章行为级建模完成**，（注意有关延迟的部分可以不看，比如forever循环，并行块，命名块等等）

学习行为级建模后，大家很有可能会认为行为级建模才是正确的，但**要记住verilog的本质是硬件描述语言而不是硬件设计语言**，因此你需要去了解行为级建模，数据流建模，结构化建模的区别，以及**为什么我们不推荐使用行为级建模**。

相关推荐学习资料：

html文档学习：[Verilog实践部分文档学习](#)（可以在这上面直接学习相关的语法——**重实践**）

推荐的自学课程：[中科大的Verilog OJ平台](#) (需要注册并登录)，推荐边看书边练手。实际部分掌握不错的**可以跳过该课程**

 **推荐学习方法：**看书作为理论学习，[Verilog实践](#)文档作为实践学习，然后就可以上手[中科大的verilog实践OJ平台](#)了，全部知识学完之后，就可以开始[拔高部分的数字设计HDLbits OJ做题网站](#)学习了。（这两个平台有一些题目是重复的，做过一遍可以跳过）

## 学习作业

1. 随时**记录自己的学习记录**，一定要如实记录，我们会不定期查看各位的学习情况。
2. [Learn C The Hard Way](#)中第**1-16节完成**，**最后将章节练习任务放在命名为“C”的文件夹中。**

**！** 这部分已经完成的同学可以选择拔高作业。

3. Verilog实践：[Verilog在线学习网站](#)——veilog学习实践，全英文（在这里推荐一个浏览器插件：**沉浸式翻译**，如果看不懂的话，就用这个插件配合学习吧！），**完成到2.4produres结束，上传截图即可**

**！** 进度快的同学继续向后做即可，做腻了可以选择拔高或选做的其他部分内容

# Getting Started

- [Getting Started](#) ○
- [Output Zero](#) ○

## Verilog Language

### Basics

- [Simple wire](#) ○
- [Four wires](#) ○
- [Inverter](#) ○
- [AND gate](#) ○
- [NOR gate](#) ○
- [XNOR gate](#) ○
- [Declaring wires](#) ○
- [7458 chip](#) ○

### Vectors

- [Vectors](#) ○
- [Vectors in more detail](#) ○
- [Vector part select](#) ○
- [Bitwise operators](#) ○
- [Four-input gates](#) ○
- [Vector concatenation operator](#) ○
- [Vector reversal 1](#) ○
- [Replication operator](#) ○
- [More replication](#) ○

### Modules: Hierarchy

- [Modules](#) ○
- [Connecting ports by position](#) ○
- [Connecting ports by name](#) ○
- [Three modules](#) ○
- [Modules and vectors](#) ○
- [Adder 1](#) ○
- [Adder 2](#) ○
- [Carry-select adder](#) ○
- [Adder-subtractor](#) ○

### Procedures

Procedures include **always**, initial, task, and function blocks. Procedures allow sequential statements (which cannot be used outside of a procedure) to be used to describe the behaviour of a circuit.

- [Always blocks \(combinational\)](#) ○
- [Always blocks \(clocked\)](#) ○
- [If statement](#) ○
- [If statement latches](#) ○
- [Case statement](#) ○
- [Priority encoder](#) ○
- [Priority encoder with casez](#) ○
- [Avoiding latches](#) ○

### Contents

- 1 Getting Started
- 2 Verilog Language
  - 2.1 Basics
  - 2.2 Vectors
  - 2.3 Modules: Hierarchy
  - 2.4 Procedures
  - 2.5 More Verilog Features
- 3 Circuits
  - 3.1 Combinational Logic
    - 3.1.1 Basic Gates
    - 3.1.2 Multiplexers
    - 3.1.3 Arithmetic Circuits
    - 3.1.4 Karnaugh Map to Circuit
  - 3.2 Sequential Logic
    - 3.2.1 Latches and Flip-Flops
    - 3.2.2 Counters
    - 3.2.3 Shift Registers
    - 3.2.4 More Circuits
    - 3.2.5 Finite State Machines
  - 3.3 Building Larger Circuits
- 4 Verification: Reading Simulations
  - 4.1 Finding bugs in code
  - 4.2 Build a circuit from a simulation waveform
- 5 Verification: Writing Testbenches
- 6 CS450

最后将以上内容放入一个文件夹里，文件夹命名为 **你的姓名-专业班级**，然后压缩为一个压缩文件即可。

请严格按照上述要求发送到邮箱：[YunDing\\_YSYX@163.com](mailto:YunDing_YSYX@163.com)

我相信这个寒假不少同学会充分利用这个黄金时间学习技术，那么以上的学习内容肯定是不够你学的，因此我准备了这个部分内容。

## 选做

# 计算机组成原理

## 地位

考研408数据结构 操作系统 **计算机组成原理** 计算机网络，计算机知识硬件部分精华科目，理工科学生必修课之一，无论考研还是工作都占据极其重要的地位，同时更是我们后续实践的基础科目。

### ? 为什么有这么重要的地位呢?

那就要知道机组具体学的内容了，以下是机组完整学习的思维导图，大家自取了解：

链接：[https://pan.baidu.com/s/1YDMg\\_OsAa2Kn5wpj7k99JA?pwd=9999](https://pan.baidu.com/s/1YDMg_OsAa2Kn5wpj7k99JA?pwd=9999) 提取码：9999

看完之后，你应该就明白为何如此重要了——这不就是我们设计芯片的基础蓝图吗？

第六七章的IO和总线部分难度大，内容多，大家在后续才会接触，因此我们寒假只需要学习完第五章处理器即可。

## 内容

以下视频的**CH3-CH4两章内容学习**，记得随手typora记录笔记和继续自己的学习记录哦！

[https://www.bilibili.com/video/BV1Ba4y1V7GD/?vd\\_source=4ec31615294fd2510d5fd40f0183648f](https://www.bilibili.com/video/BV1Ba4y1V7GD/?vd_source=4ec31615294fd2510d5fd40f0183648f)

机组部分学习会大量涉及你学过的**数电**知识，如果当时没有好好学，那就要比较吃力了。

！ 注意！该课程是使用教材为**MIPS指令集版本**，我们之后设计芯片，使用的是**RISCV指令集版本**，但是考虑到该课程体系完善，效果好，指令集思想是相通的，因此我们仍然选择了该视频课程，配套书籍为黑皮书：计算机组成与设计：硬件软件接口，图示如下（群内有电子版资料），想搭配书籍的同学可以参考——不推荐纯看书，黑皮书阅读难度大，知识点深而且广，很难理清知识点。

因此，也可以选择你觉得好的计组课程或者硬啃黑皮书，学到知识就好！



## 计算机组成与设计：硬件软件接口（原书第5版）

戴维 A. 帕特森 (David A. Patterson), 约翰 L. 亨尼斯 (John L. Hennessy)

☆ 5.0 / 5.0 9 comments

《计算机组成与设计：硬件/软件接口》是计算机组成与设计的经典畅销教材，第5版经过全面更新，关注后PC时代发生在计算机体系结构领域的革命性变革——从单核处理器到多核微处理器，从串行到并行。本书特别关注移动计算和云计算，通过平板电脑、云体系结构以及ARM（移动计算设备）和x86（云计算）体系结构来探索和揭示这场技术变革。

与前几版一样，本书采用MIPS处理器讲解计算机硬件技术、汇编语言、计算机算术、流水线、存储器层次结构以及I/O等基本功能。

种类: Computers  
年: 2015  
出版社: 机械工业出版社  
页: 557  
ISBN 13: 9787111504825  
文件: PDF, 153.37 MB

Content Type: 书籍  
出版: 5th  
语言: Chinese  
ISBN 10: 7111504828  
系列: 计算机科学丛书  
IPFS: CID, CID Blake2b

▶ 在线阅读

↓ 下载 (PDF, 153.37 MB)

发送到

平装

书籍有问题吗?



### 学完机组我能干什么？

1. 为之后的数字设计，硬件设计，芯片设计等等相关方面打基础，缺少机组知识你将寸步难行
2. 对考研有很大的帮助
3. 硬件相关学校课程你的修读难度会很低

## 任务

交回相关markdown笔记（我们推荐大家进行**总结归纳和心得记录**，这真的很有帮助），以此说明你完成了此部分的学习。

## 拔高

### 1. 数字设计

数字设计重要性不言而喻，因此拔高部分你可以选择继续学习数字设计，学的越多越好，**书籍内容一直学到行为级建模结束**，过程中进行实践，学习；[Verilog在线学习网站](#)中**一二三四章**都非常推荐学习，所以加油吧！



### 那我机组部分的实践内容嘞？

机组的学习内容就决定，它**学到的很多是思想**，是**整体的架构**，这就意味**机组实践内容需要你技术栈更全更深**，因此机组实践内容还要向后放，简单部分实践在之后会有一部分，处理器设计方面内容则需要寒假开学之后，视个人进度逐步进行实践。



## 2. Learn C the hard way

LCTHW: <https://wizardforcel.gitbooks.io/lcthw/content/preface.html> 其中的**26、37-41、43、45-47不需要学习**，性价比比较低，不推荐学习，**其他内容我们都很推荐学习**，想要技术很强的话，都可以试着去学习，并在其中锻炼自己gdb等debug工具使用和相关能力思维。

其中有关“一生一芯”的必须完成部分如下：

### 📌 C语言入门必做题

Learn C the hard way 📌 练习0到练习18，练习32，练习33，练习42，练习44 需要完成编程算法以及附加题.

- 这是入学答辩的一部分，如果你在有些地方被卡住，可以通过RTFW或咨询助教
- 参考资料使用上文提到的学习资料

## 3. 一生一芯课程PA

PA是我们后续学习中非常重要的一部分内容，目前我们已经把PA0相关的基础知识进行了补全，大家**可以去尝试PA0的相关内容学习**，不过想开PA1还需要一些时间，PA1的内容需要数据结构，Lcthw都学到不错的地步并具备一定的编程思想，然后就需要你的时间和精力花费了。

！ 说这么多，其实就是告诉大家**PA1部分的学习内容难度很高**，可能会花费不少时间，同时这也是后续预学习答辩的重要内容之一，所以如果**你想挑战自己的能力极限**，那现在去做PA1也是可以的，加油！

以上就是拔高作业的全部内容，说实话如果你能在这个寒假把以上内容全做完，意味着——》寒假结束后一个月内即可参加预学习答辩并顺利通过，正式进入“一生一芯”项目，后续继续保持学习，那么你的技术将比我更强！暑假你完全可以凭借自己的实力进入线下基地。**但是也请大家劳逸结合，考虑自己的能力和精力，合理学习，不要为了求快，去燃烧自己的热情与生命，这样是非常得不偿失的，**

？ 难度这么高几乎无法完成，那拔高作业布置这么多是为什么？

1. 想让大家明白自己要学的内容还有很多。
2. 让大家可以自由选择拔高的学习内容，以上三项，皆可自由选择学习，不存在非常强烈的关联先后关系。

3. 并没有要求大家拔高作业全部完成，而是根据自己情况，可以完成多少就完成多少。

本作品《"太理工一生一芯工作室前置讲义寒假研学篇"》由 许鹏远 创作，并采用 CC BY-SA 4.0 协议进行授权。

遵循CC BY-SA 4.0开源协议：<https://creativecommons.org/licenses/by-sa/4.0/deed.en>

转载或使用请标注所有者：许鹏远，太理 “一生一芯” 工作室