

2020 秋季计组课设预习建议

一、计组课程概述与 cscore 平台介绍

计算机组成包含理论课与课程设计两部分：理论课进行原理讲解，课程设计将最终利用相关知识完成五阶段流水线的、支持异常中断、可在 FPGA 上运行的 MIPS CPU。为了实现这一目标，需要经过基础知识部分、CPU 部件等前期内容的学习。

为了帮助大家对于计组课程设计（以下简称“课设”）的内容和流程有一个全局的认识，本文给出了课设各部分的内容概览、学习目标，同时对课设使用的在线学习平台——[Cscore](#) 进行了初步讲解。

1. 计组理论课简述

课程讲授数字电路的基本知识与计算机各组成部分的理论知识。

2. 计组课设简述

课设的组织形式为闯关制、请完整学习 Cscore 网站 [《Hello! Computer Organization!》](#) 章节内容以进一步了解课程运行，该章节内容包括课程体系及课程运行流程简介、SPOC 平台使用介绍、实验评分细则、学术不端惩罚措施、讨论区规范、助教团队介绍以及 Entrance Quiz 等内容。

1) 教程部分

在开始搭建 [CPU](#) 以前，大家需要先学习[数制](#)、[逻辑电路基础知识](#)、[Logisim](#)、[Verilog-HDL 与 ISE](#)、[MIPS 指令集及汇编语言](#) 这些部分。他们依次分别对应 Cscore 平台的[数制](#)、[逻辑电路基础知识](#)、[Logisim](#)、[Verilog-HDL 与 ISE](#)、[MIPS 指令集及汇编语言](#)。具体的学习目标和内容介绍将在后文的[教程部分概览](#) 进行描述，方便大家进行暑期预习。

课设第一次课上考试为教程部分的测试，即 **Pre 测试**。在这一时间节点之前，同学们须完成教程部分的学习。课设秋季运行的 P0-P2，随教学周推进，配合理论授课进一步强化各项工具使用、数字电路、汇编语言及 Verilog HDL 等基础知识的掌握。

2) CPU 部分

在完成了教程部分的学习后，学有余力的同学可以根据个人学习情况进行 CPU 原理和实现方法的预习，包括对[CPU 部件](#)、[单周期 CPU](#)、[流水线 CPU](#) 进行了解和学习。这三个内容对应课设 P3-P6。

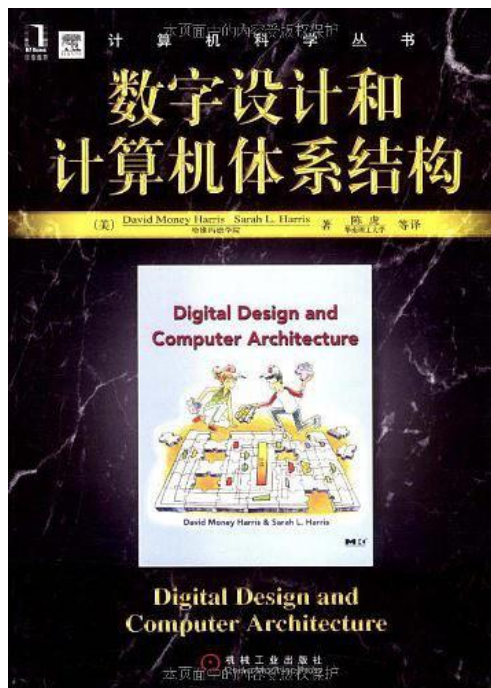
在课程后期，CPU 将要增加“支持异常中断”、“在 FPGA 上运行”等要求。这一部分内容可以查看 [See MIPS Run Linux](#) 进行提前了解。

3. Cscore 平台简介

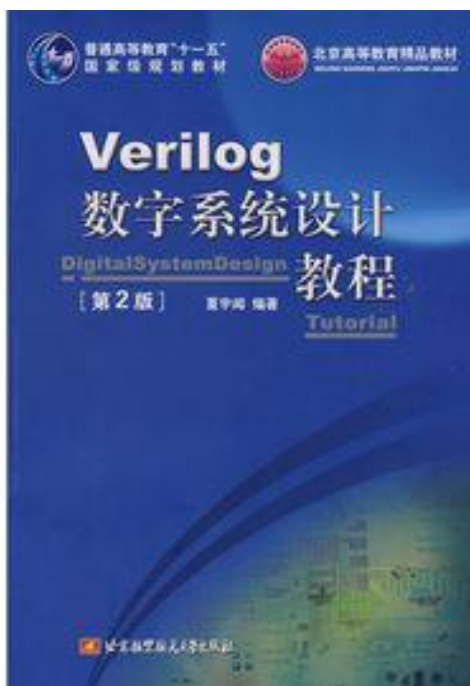
Cscore 平台是课设教学、测试的平台。大家需要阅读上面的教程、完成题目，并在课上测试中在该平台上完成测试内容的提交。Cscore 的注册流程请查看 [Cscore 注册流程](#)，使用方法请参考 [Cscore 使用参考](#)。

4. 计组课设教材

计组课设的教材有两本，分别为：



以下简记为黑书



以下简记为蓝书

5. 计组课设重要资料链接

1) 教材 pdf 及往届理论课课件：

链接：<https://pan.baidu.com/s/1aHy1MZfPMXRMIh3Md3tFfg>

提取码：249m

2) Verilog 自学+评测网站推荐

https://hdlbits.01xz.net/wiki/Main_Page

3) MIPS 中英文指令集

中文：http://cscore.net.cn/assets/courseware/v1/7a73e4d815b91546e12330bd139dfda1/asset-v1:BUAA+B3I062410+2017_Summer+type@asset+block/asset-v1_BUAA_B3I062410_2016_T1_type_asset_block_MIPS-C%E6%8C%87%E4%BB%A4%E9%9B%86_%E6%A0%A1%E5%AF%B9%E5%AE%8C%E6%88%90%E7%89%88_-%E6%8C%87%E4%BB%A4%E6%8E%92%E5%BA%8F.pdf

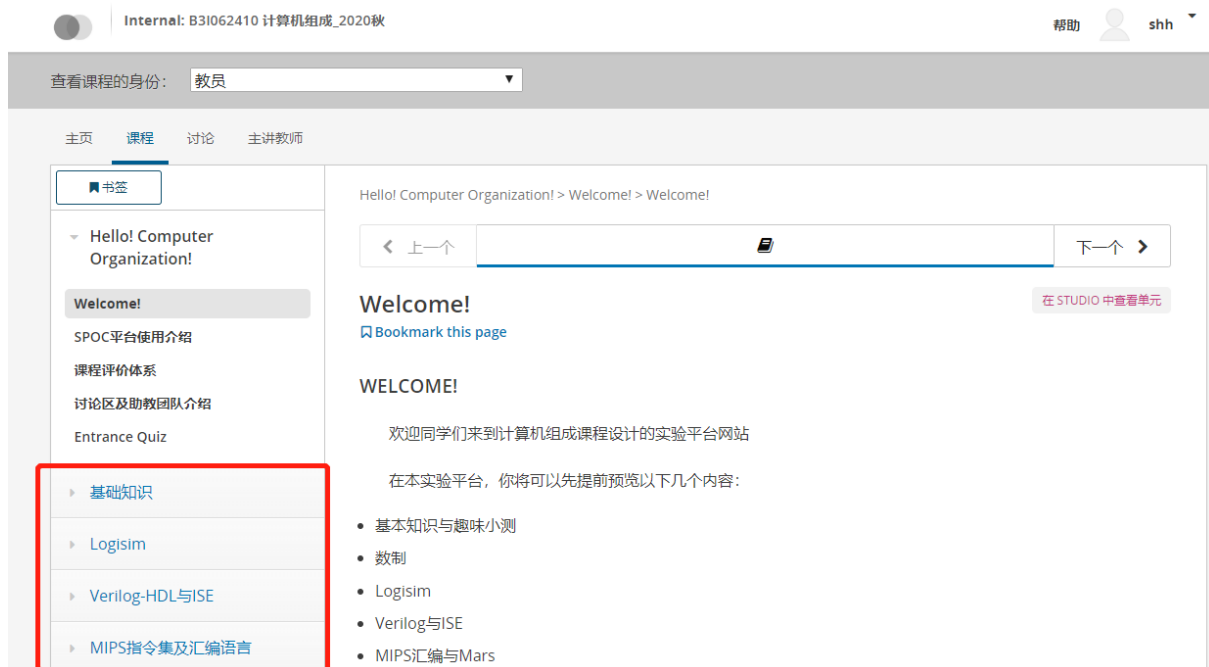
英文: http://cscore.net.cn/assets/courseware/v1/9f6c8aec05165d9419761197778a5225/asset-v1:BUAA+B3I062410+2017_Summer+type@asset+block/MIPS_Vol2_%E6%8C%87%E4%BB%A4%E9%9B%86_.pdf

4) See MIPS Run Linux

http://studio.cscore.net.cn/assets/courseware/v1/98dccbb370200781975f09a7ae7cba4f/asset-v1:BUAA+B3I062410+2017_Summer+type@asset+block/asset-v1BUAA_B3I062410_2016_T1_type_asset_block_See_MIPS_Run_Linux.pdf

二、教程部分概览

教程部分的具体内容可以在 [Cscore 平台](#) 查看，故这里只做简单的描述和概括性的说明。



1. 数制、逻辑电路基础知识、Logisim

这一部分的具体教程为 Cscore 平台的[数制](#)、[逻辑电路基础知识](#)、[Logisim](#)，对应**黑书第 1-3 章**。

将从二进制出发，介绍二进制的原理，从而进一步引出门电路、状态机这两个重要的数字逻辑电路的基础知识。在对门电路组成的[组合逻辑电路](#)、门电路与状态机共同组成的[时序逻辑电路](#)有了基础认识后，将使用 Logisim 工具搭建数字电路，以完成特定的功能，并在 Cscore 平台上进行自动测试。Logisim 的下载、使用方法请参考 [Logisim 相关资料](#)。

2. Verilog-HDL 与 ISE

这一部分的具体教程为 Cscore 平台的[Verilog-HDL 与 ISE](#)，对应**黑书第 4 章**和**蓝书全部**。

Logisim 可以实现简单的硬件设计，但对于复杂的硬件无能为力。硬件描述语言（HDL）以代码的形式对硬件进行描述，可以利用循环、条件判断等功能对硬件进行建模，是课设搭建 CPU 的主要方式。

Verilog 是一种 HDL，ISE 是用来开发、调试 Verilog 工程的集成开发环境。它们是大家在教程部分须掌握的语言/工具。

硬件设计不同于软件，需要实际部署到设备上才能够进一步判断是否符合设计要求。仿真工具是在虚拟的环境中对设计的硬件行为进行模拟，从而在软件环境辅助硬件调试。ISE 自带的 ISim 工具便是一款仿真工具，帮助进行 Verilog 工程的调试。

3. MIPS 指令集及汇编语言

这一部分的具体教程为 Cscore 平台的 [MIPS 指令集及汇编语言](#)，对应**黑书第 6 章**。

一台计算机大体可以分为硬件和软件两大部分。硬件与软件之间的界面是指令集。指令集由几十条至几百条功能各异的指令组成。每条指令都定义了一个独立功能，如加法运算等。从软件的角度，无论多复杂的软件都是由这些指令组成的集合。从硬件的角度，每条指令定义的具体功能则由硬件负责执行完成。为了有效支持高级程序设计语言，MIPS 定义了加载和存储指令、算数运算指令、逻辑运算指令、比较指令、分支指令、跳转指令等指令。除了上述常规指令外，MIPS 还定义了一些与特定体系结构相关的指令，如协处理器、Cache、MMU 管理等指令。

每条机器指令实际对应了由相应规则定义的 01 串，为了方便表示各类 01 串的功能（包括加载、存储、寄存器-寄存器运算、寄存器-立即数运算、分支、跳转等），人们发明了助记符来简化表示“指令”，如将 01 串中**特定位置**出现的“001001”简记为“addiu”。addiu 指令的功能是将源寄存器与立即数相加并将结果写入目的寄存器。其中，源寄存器、立即数以及目的寄存器分别由 01 串中某部分位置的值指示。

在课设中，我们将使用 MIPS 汇编程序对 CPU 进行功能测试，但直接编写机器指令（二进制编码的 01 串）过于低效。我们将借助 MARS 这款汇编器，将由助记符编写的汇编程序汇编为 MIPS CPU 可执行的二进制编码的机器指令，具体介绍请参考 [MARS 的获取与使用](#)。

此外，MARS 可以模拟 CPU 执行汇编指令时，相应寄存器、存储等部件的变化情况。在[教程部分](#)，大家要学习 [MIPS 指令集](#)、学习利用 MIPS 指令集编写达成特定功能的汇编程序，并通过 MARS 进行仿真，查看程序是否符合期望，同时需要具备基于 Mars 调试汇编代码的能力。在[CPU 部分](#)，大家要使用 MIPS 指令集编写测试用的汇编程序，通过对比自己的 CPU 与 MARS 的执行结果来对 CPU 进行测试。

在课设的 P8（即最后一关），CPU 此时已经相对完整，需要在自己的 CPU 上执行特定的汇编程序，来实现更高级的功能（如中断处理、点亮 LED、串口通信）等，需要对 MIPS 编程进行进一步应用。

三、CPU 部分概览

由于内容涉及知识较多，这一部分在 Cscore 平台尚未开放。请有兴趣的同学结合教材对应章节和此文档，对相应理论知识进行学习，使用 Logisim 或 Verilog 搭建自己的 CPU。

1. CPU 部件

这一部分对应**黑书第 5 章**。

CPU 工程与前面其他预习内容最本质区别就是其**多模块协作**的特点。很多同学在学习 C 语言时都经历过“一 main 到底式”编程，这种单一面向过程执行的方法对于 CPU 这样量级的工程来说显然是不适用的，大家根本无法在一个全局环境下照顾到各处细节，即使做到了，也会遍布 Bug。

面对复杂的 CPU 结构，首先要对其进行多模块“解剖”，看看书中都介绍了哪些必要的模块，大致了解各个模块是如何由数据通路进行连接。然后可以开始对模块逐一剖析，具体学习它们的功能。在设计一个模块时，暂时不考虑它在整体中处于什么样的位置，与其他模块会如何作用。大家可以

将这个过程看作单一简单部件的设计。最后再将所有的模块统筹在一起，连接到顶层的电路中，对于一些细节的地方可以进行微调。

这个设计可以看作是一个“整体—部分—整体”的过程，好比组装一款模型，先看一眼成品效果图，然后去完成各个零部件，最后把这些部件进行拼接。

2. 单周期 CPU

这一部分对应**黑书第 7 章**。

1) Logisim 实现单周期 CPU

为了大家更好的理解，CPU 部分的实验（P3 及之后）也会从 Logisim 搭建单周期 CPU 开始，支持的指令集可以参考黑书或 MIPS 中文指令集，具体步骤不在此赘述。

Logisim 搭建的 CPU 使用 Logisim 中的 ROM 部件模拟 CPU 中的指令存储器模块。因此，如果大家想要对自己搭建的 CPU 进行测试，要进行如下几点操作：

- i. 将 MARS 中的 Settings — Memory Configuration 设置为 Compact Data at Address 0。
- ii. 将 MARS 中的 Settings — Delayed branching 取消勾选。
- iii. 编写好汇编程序后，使用 MARS 中 File — Dump Memory 将指令的机器码导出。其中 Memory Segment 选择“.text”，Dump Format 选择“Hexadecimal Text”。
- iv. 在导出的文件的第一行加入 Logisim 所需的识别文字“v2.0 raw”，右键 ROM 部件，选择 Load Image 即可将编写的机器码加载。
- v. 需要注意的是，MARS 中的指令地址是从 0x0000_3000 开始的，而导入 ROM 后地址从 0 开始，要对其做一些处理才可以使 CPU 正常运转。

2) Verilog 实现单周期 CPU

搭建过 Logisim 版本 CPU 后，大家可以尝试从具体到抽象的过渡，也就是将 Logisim 中的电路元件转化为 Verilog HDL 描述，每个模块可以对应到一个 Verilog 源文件中。如果大家对 Verilog 已经足够熟悉，这个过程会进行得非常顺利。

测试 Verilog 版的单周期 CPU 工程，可以使用 Verilog 的系统任务 \$readmemh 读取 MARS 导出的十六进制指令机器码，然后使用仿真功能进行调试。如果想观察中间寄存器堆和内存的变化，可以观察 ISim 仿真波形，或使用系统任务 \$display 将信息输出到控制台。

3. 流水线 CPU

这一部分对应**黑书第 7 章**。

从单周期 CPU 转化至流水线 CPU，采用了一种“分级”的思想。单周期 CPU 是在一个时钟周期执行完一条指令，而流水线则将单周期的执行过程拆分为五部分（计组实验采用的是五级流水线），一个周期只让一条指令向前前进一级，因而流水线中可以同时存在多条并行执行的指令。微观来看单条指令，执行时间似乎变长，但在多条指令并行的情况下，每一级的逻辑复杂度相比单周期 CPU 都有所降低，所以流水线技术宏观上可以极大地提高 CPU 的主频。但是由于流水线并行执行不同指令，会在某些情况下出现名为“冒险”的现象，这一现象的原理与解决方法将在 P5 被用到。

大家可以先从不考虑冒险的流水线入手，搭建流水线 CPU 的数据通路。既然要让指令能够在流水线中停留多个周期，就必然要引入多个流水级寄存器来存储相关的信息。大家可以从这个角度去设计。测试时需要开启 MARS 的 Delayed branching。

在测试中大家会发现有时 CPU 写入的数据与期望的数据并不相同，满足条件时却并没有分支到指定的 PC 地址。这些并不一定意味着大家对指令的数据通路设计有误，而更可能是出现了冒险的情况。大家可以阅读**黑书第 7 章**有关冒险的内容进行尝试处理，也可以在秋季开学后查看 Cscore 相关教程进行学习、应用。

四、 其他参考与扩展

1. Verilog 自学、自测网站——HDLbits

在进行 Verilog 入门的过程中，可以使用 HDLbits 网站 https://hdlbits.01xz.net/wiki/Main_Page 进行简单的仿真、波形图观看、刷题。值得注意的是，推荐大家在入门后，使用 ISE 和 ISim 进行仿真、调试，仿真结果以 ISim 为准。

2. 流水线 CPU 的冒险

除了之前提到的两本官方教材外，大家仍需要学习 Cscore 平台的 MIPS-C 指令集（英/中文版），了解指令的具体行为和格式，为搭建 CPU 做准备。在进行流水线的“冒险”、“延迟槽”处理时，需要参考英文版指令集中对于 PC 变化的描述。

3. See MIPS Run Linux

大家还可以根据自身情况，提前阅读《See MIPS Run Linux》这一本书。从计组到 OS，至以后涉及到系统相关的项目，将都会有这本书的身影。在 P7 及之后，这将成为大家主要的参考书。