#2018.10.15~2019.8.8 NSCSCC(龙芯杯)

从去年 10 月到今年 8 月，将近 10 个月时间的准备，终于告一段落，很遗憾没能成功打进决赛。

说是有 10 个月的时间准备，但实际花在这个比赛上面的时间可能也就 1 个月左右。比赛对我们来说也算是完结了，也该花点时间稍微总结一下这十个月来的一些心得和体会。

大三上学期——初试单周期

第一次知晓这个比赛是当时辅导员在通知群中发布了一条消息，就是关于这个比赛的 2019 年招新。然后消息中有关于大赛官网的链接，我当时简单的浏览了一下官网，但是并不清楚这个比赛究竟是干什么的，但是从这个比赛的名字也可以知晓，这是一个跟计算机系统相关的赛事。我当时觉得，现在都已经大三了，可能往后也没什么比赛能够参加了，而且自己到现在为止并没有参加过一个比较有规格的计算机方向的比赛，便决定去了解一下这个比赛。之后也就去参加的这个招新会。

依稀记得，陆老师在召集大三学生参赛的那一天，大约 20 来个人在机房开会，上一届学长简要把比赛给我们介绍一通，之后陆老师和清华大学的向勇老师也跟我们聊了一下，同时也向我们询问了参赛的目的。我当时记得，我说参加这个比赛是想在大学期间能够参加一个有分量的全国性比赛，让大学至少不留下什么遗憾。到现在，我觉得可以再加上一条原因，兴趣使然。之后就开始了对比赛的准备，我们约定还是沿袭上一届传统，每两周开一次会，由此督促学习。

通过这个招新会，也大致知道了这个比赛要做什么，简单的说就是用硬件编程语言写一个 CPU,然后能够运行一些简单软件甚至操作系统在这个 CPU 上。这之后也更对这个比赛有点兴趣了，因为之前自己便对硬件这一块比较感兴趣，尤其是比较希望了解到一个计算机系统的硬件是如何工作的。而这个比赛正与我的兴趣不谋而合，当时便下定决心一定要把这个比赛坚持做完即便挂科。

做这个比赛，最开始比较烦的就是对于开发环境的搭建，又是 Linux 又是 Vivado 的，很多东西都是第一次听说，但是我自己对这些东西稍微知道一点，搭起来并不是什么难事。一个主要问题就是电脑硬盘空间不够，Vivado 加上 Linux 环境至少需要留出 50G 的空间，我的电脑总共只有 256G 的 SSD，装上很多软件之后空间并不够，无奈之下就卸载了一部分不常用的软件才把这个环境给搭好。

两周后的周末第一次开会，感觉人只有十来个吧，没有招新会来的多。第一次开会，大家也没有做什么，仅仅装了开发环境，一些人开发环境甚至没有装好，然后学长开始向我们展示他们上一届的作品。到现在，我是记不清他们当

时展示了什么，只记得是讲 bit 文件烧到板中，然后板子就开始工作了。再然后就开始跟我们介绍之后要做的一些任务。

此处插一句，相比于上一届，我个人觉得我本身有一件事做的比较差，就是对于每两周一次的简单总结。以至于到现在我在写这个文档的时候不知道写什么，也不知道当时在某一时间段究竟做了什么。因此这件事还是有坚持做下去的必要的，应该把他当作一种习惯，这个东西是对某一时间段的一个总结回顾，日后可能会有时需要去回顾或者去参考过往的经验教训。现如今，倒是有时会做一些记录，但是做的还是不够好，不够彻底。

大三上学期是大家还在各自学习的阶段，学习 Verilog 语法，学习 MIPS 架构，了解比赛规则等等。从一个只有 7 条指令的单周期 CPU 开始搭建，逐步增加指令，R 型，I 型，J 型指令选择有代表性的一些指令都实现一下。说实话，对于一个 Verilog 都不怎么会的人来说，上手一个单周期也是及其困难的，而且当时计组、体系结构什么的都没有学，像大小端这种比较简单的概念可能都难以区分。

实现单周期的目的不是说要使用单周期这种架构来做一个比较完备的 CPU，而是通过写一个简单的单周期 CPU 来渐渐学习 MIPS 架构、Verilog 语法，了解一些比较常见的名词概念。通过单周期，可以知道条指令的执行需要经过哪些阶段，若干个模块如何协调工作。

在写单周期的过程中，问题还是很多的，参考老师给的 7 条指令单周期的代码，很多地方不明白为什么要这么设计，但是通过查阅 MIPS 手册，对整个代码进行一个全局的了解，慢慢的很多地方也是可以搞定的。再不能搞懂的地方就是去问学长了，然后学长就会给解释清楚。我在设计单周期的 CPU 的时候还参考了《自己动手写 CPU》这本书，当时还不知道这本书是用五级流水线的架构来写的（当时也不知道流水线是什么），然后就参照他这样分为五个阶段，取值译码，执行，访存，回写，把这五个阶段整合到单周期的设计中，也就是说，在单周期的设计里依旧是把指令的执行按照这样的一个处理过程来进行，只不过都放在了一个周期里。这样在后面不断的添加指令的过程中，只需要添加一些代码，整体架构不用改动，除非遇到一些比较“怪异的指令”。

就这样，在不断的修修改改，学习架构，学习语法的过程中，逐渐掌握的 Verilog 的基本语法以及他的开发的一些基本套路，还有对 MIPS 架构也是有了一定的了解，对于 MIPS 指令如何实现能够了然于胸。由于平时还有课程，所以，在比赛这一块花的时间不是很多，进展有些慢，到第 12 周左右（本学期过了一大半了）才大致确定单周期的任务可以暂时停止，再写意义不大，可以进行下一步的进阶任务。

但其实到现在来反过头来看这个进度，还是很慢的，大半个学期才仅仅完成到写单周期的任务，对于流水线这些概念还是完全无知的。到下学期，完成流水线时花的时间可能会更多，同时还要加上 cp0 寄存器。整体架构要改很多次，这些都是一个没有经验的人来写的必然经过，所以会花费很多时间来调整，这样到完成流水线之后，留给优化的时间就不多了，这也是导致我们这届性能上

不去的一个原因——优化 CPU 时间太少。要如何少走弯路，从一开始就能有一个比较好的架构，还是要多看别人的设计，多参考一些书籍，对整体进行一个核心把控，小组成员内多多讨论，完整确定架构之后再来写代码。

到 2018 年底能来按时参加组会的也没有几人了。可能不到 10 个人，所以说句实在话，这个比赛还是很劝退的，难度也是有点大，对于这个比赛，如果完全是抱着想比赛获奖这个心态来的，那么对不起，这个比赛只会越来越难，能够进入决赛的机会是很小的，而且二等奖以上基本被其他几所学校长期“霸占”，要拿奖也就三等奖这个水平的样子。

然后在 11 月份的时候，我们还剩的一些人大致分成了两个小组，小组成员之间互相督促互相学习（感觉没什么必要，反正每个人都是要对整体进行学习掌握的）。大概是月末，我最终参加初赛的队友——范志鹏也加入到这个比赛了，他是电信学院的。从他的介绍中，感觉他以前是做过一些类似的开发，感觉很强的样子（确实很强）。他临时安排在我们小组。再然后就陆陆续续的有考试了，花在比赛上的时间也就少了一些，偶尔看看书，写两行代码，没什么太大进度。

在寒假期间是计划要实现一下多周期和流水线，但是在家的效率极其低，冬天冷得都不想伸爪更别说敲代码，寒假的时间也没怎么利用好。开学之后用了一个星期把多周期简单写了一下，实现若干条指令，就不管了，毕竟最后关键是流水线。

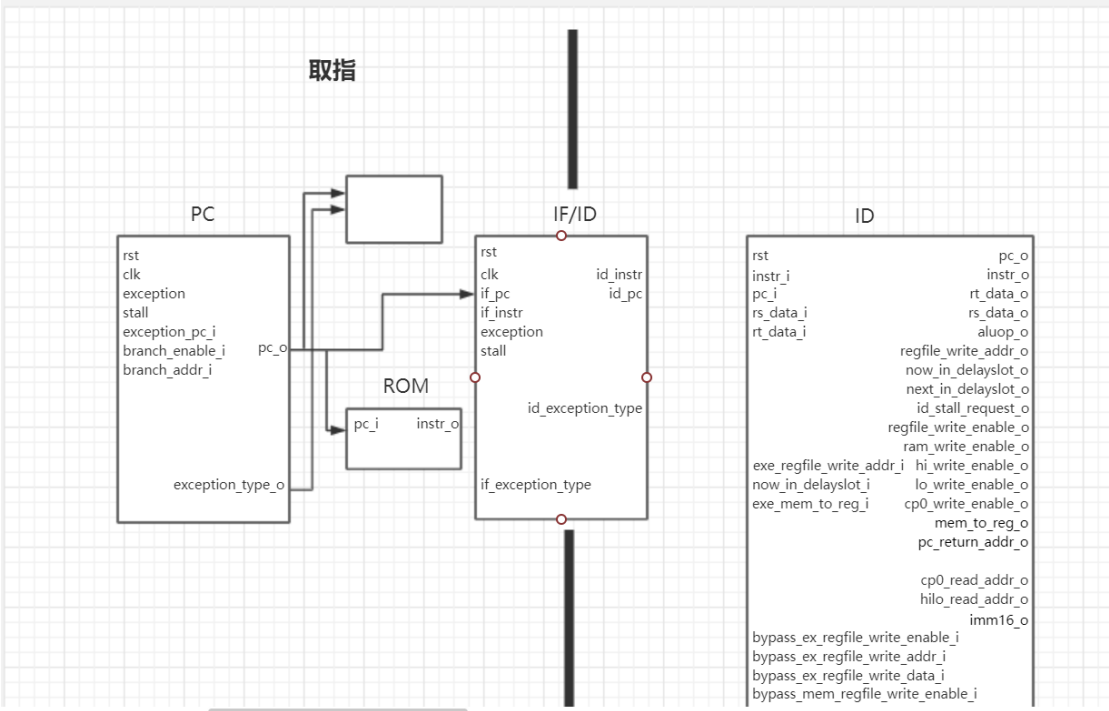
而此时参赛人更少了，后来陆老师又找到几个人参赛，到最后初赛也就能够凑齐两队人（6 个人，一队 2 人一队 4 人），我和范志鹏一组。

大三下学期——流水线的挣扎

大三下的关键是学习流水线，并实现 MIPS 五级流水线，但这个时候 CA 还没学，流水线是什么也不知道，只能自己去学。此时也组好了队，我跟范志鹏一队，我俩计划都各自先实现一个简单流水线，以对流水线有个认识，之后一起进行整体架构设计并一起写代码，通过 git 和 github 管理代码。

在 4 月之前，我们俩基本上都是在各自写各自的流水线，以对整个流水线的架构有一定的认识与了解。这样做好处还是很多的，当我俩都对整个流水线有了认识只会，我们在共同设计的过程中，能够发现彼此设计不好的地方，比如队友来设计一个 if 模块，他把接口都写好只会我可以来进行复核，因为我也了解这个模块的设计，我可清楚的知道他要这些接口的目的，也可以帮他查找错误接口设计。这样便可以免去让他跟我介绍这个接口这样繁琐的工作。只会在写代码的过程中也是有所帮助的，当队友在写一些模块的时候，写了一些 bug，而他自己很难发现这个 bug，我本身了解这个模块该怎么写的时候查看他的代码就很容易找到 bug 位置（给他人找错每个人还是很擅长的）。在加了大概几十条指令之后，我和范志鹏决定开始协调工作。

首先就是要来设计一个整体的架构。设计架构的时候需要把架构图画出来，每个模块的接口都相应的标明。我们是在 processon.com 这个网站上做这个工作的。这个网站提供了一个在线做图可多人协作的应用，十分方便在上面做各种图，还可以多人协作实时查看，十分方便，这样我两即使不面对面也能一起花设计图纸。示例如下：



架构是在不断的进行调整，因为每加上一类指令，都需要对整体结果进行相关调整，从 IF 模块到 WB 模块，来来回回改，不知道改了多少次。在前面加普通指令的过程中还是问题不大的，比较好加，仅仅需要稍微改改译码模块和执行模块等。后来加 cp0 的时候，感觉还是很烦人的。因为跟 cp0 相关的一些指令需要整个流水线也做出相关调整，同时还有异常处理过程也需要跟整个流水线相关。

其实，当真正地把整个架构设计好之后，写代码是比较快的，主要是代码写好之后 debug 的过程是比较漫长的，还好大赛官方给了一个 trace 对比的功能，这也减轻了 debug 的负担。

最终在考试周之前算是把大赛要求的 57 条指令以五级流水的架构实现了，通过了功能仿真，并且能够下板运行。但性能分低的可怜只有 0 点几分，频率也只有 80+M。我们庆幸仿真一过，下板也就过了，下板并没有出错，毕竟下板出错进行 ILA 的过程是及其让人头疼的。此时也到了各个课程结束并陆续有些考试和大作业的时间了，比赛的进度再次放缓下来。我们计划等过完考试周对 CPU 进行优化，即增加 cache 等相关工作。但是计划是计划，现实是现实，cache 的增加也是一波三折。

等过完考试周，在暑假期间便是给 CPU 加 cache 由此提高性能。找了一个时间，我跟范志鹏商量着如何设计 cache,由于他是电信学院的，CA 没有学过，导致他最开始对于 cache 的组相联的理解是错误的，后来我给他找了我们上课 PPT，给他讲解了一番。最终我俩也是决定先写一个简单的四路组相联的 cache 试试看，采用 fifo 替换策略，指令 cache 和数据 cache 各 1Kb。设计好之后，范志鹏就开始写了，是他把这个 cache 的主体代码进行了编写，我后来仅仅帮他 debug 了一下或增加删改了一些代码。同时我在这段时间学习 tlb 相关内容。

指令 cache 他用了很短的时间便写好了，并且可以正常工作，把这个 cache 启用之后，性能的确快了不少。但是之后的数据 cache 便有些艰难，写好之后怎么调都是会有一些 bug，而且 bug 他也找不出来。此时，我由于身体原因，回家养病了两周，在家我稍微抽出了一点时间帮他 debug,然后就发现了（果然很擅长给别人找错）。这些 bug 也不是很复杂的一些 bug，就是一些很简单但容易出错的地方，一不留神就会写错。同时，在没有找到这些 bug 之前，我们对 cache 的添加我们是做了两手准备，一个是使用 IP 核，一个是自己写。使用 IP 核只需调一下 AXI 接口就好，相对比较简单，所以很快就能正常工作，但是用 IP 核的话性能并不能提高很多，因为 system cache 与 CPU 核的连接是通过 AXI 协议的，而 AXI 协议有握手是很慢的，所以以后要想能有高性能 CPU 还是得自己写 cache。但是自己写就需要花大功夫了，全相联还是组相联还是直接映射？替换策略算是 FIFO 还是 LRU 还是随机抑或其他？cache 的大小如何确定，组内大小如何确定等等都是 cache 的设计要点。

cache 的优化空间还是很大的，我们自己写的 CPU 并不是很好，都是使用的最简单的一些策略。最开始我们是用的 1kb 指令 cache,后来增加到 2kb，发现性能并没有提高太多？？于是我们便不再对 cache 进行扩容，后来在大赛交流群里听别的大佬说大概 4kB 指令 cache 和 4kb 数据 cache 的性能会比较好。同时，我们这个 cache 直接用 LUT 综合出来的，比较浪费资源，而且使用大量 LUT 资源之后综合过程巨慢。所以最好还是要用 bram。而且 CPU 与 cache 的接口设计的也不是特别好，CPU 发出数据请求到从 cache 直接返回数据还是需要三个周期，最终导致我们性能分只有 14.0，88M 频率。不过也没办法，整个数据时间并不多，我生病期间基本没“输出”，范志鹏在华为实习工作日白天基本没时间就周末或晚上做一做，如果时间充足，确实可以重新设计，进行更好的优化。在性能这一块，最应该花时间的也就是 cache 这一块了，也只有 cache 设计好了才能把整体性能分提上去。同时另外一个队采用的是 IP 核，他们的性能分只有 7 左右，但是他们的频率可以调到 110MHz,这点我觉得还是很厉害的，我们的 CPU 最终也只有 88MHz,我也还是挺好奇他们如何做到 110M 的。

就这样，到 8 月 5 号提交初赛作品，我们也只有这么个东西来提交了，个人感觉这个性能分也进不了决赛。因为从大赛交流群可以看出来好多队都是跑了 20 分以上的性能分。虽然去年进入决赛的最低分是 0.8，但是这个比赛就是这样，一届比一届强，性能分会不断的往上加。各个学校参赛队肯定会参考上一届的代码，不断优化。所以今年进入决赛最低分为 24 也不奇怪。也就是说，性能提高了 30 倍！

预赛结果的通知

各参赛队：

非常感谢大家积极参加“第三届全国大学生计算机系统能力培养大赛（龙芯杯）”并及时提交预赛作品。本次大赛预赛作品的评审规则如下：

1. 评审只关注功能分和性能分，其他功能展示不在预赛评审范围内；
2. 性能测试程序未通过的，其与 GS132 的实际运行时间比值按 0.1 计算；

经过我们评审组的复审，以及大赛组委会的审核，确定了入围决赛的 21 支队伍。

决赛入围名单如下：

	入围决赛的参赛队			预赛分数		
序号	队伍名称	学校	参赛学生	功能分	性能分	频率 MHz
1	真香队	中国科学院大学	李奉治、王华强、刘蕴哲、陈欲晓	100.000	71.441	110
2	毕业就去送快递	北京邮电大学	于海鑫、盛熙铭、尹思维	100.000	61.376	100
3	Hypothetic CPU	哈尔滨工业大学（威海）	柴可、戴天宇、胡起、胡森	100.000	60.256	120
4	HITSZers	哈尔滨工业大学（深圳）	胡博涵、施杨、陈泓佚	100.000	52.652	109
5	编程是一件很危险的事情	清华大学	陈晟祺、周聿浩、刘晓义	100.000	52.157	80
6	天骄四连	南京理工大学	孔昊、李美玲、谢城、王鹏浩	100.000	48.565	89
7	ECNUCoder	华东师范大学	李一鸣、李明松、何云杰、陶琦琦	100.000	47.099	80
8	河北大学 2 队	河北大学	蔡纪良、韩思元、李宝奎、韩晓峰	100.000	46.18	110
9	北京航空航天大学 2 队	北京航空航天大学	陆俊峰、马振亚、孟繁瑞、张朝阳	100.000	43.864	82
10	河北大学 1 队	河北大学	吴建、张宏伟、李明浩、潘银	100.000	43.468	83
11	西工大 1 队	西北工业大学	蔺嘉炜、严盛恢、杨冰	100.000	39.92	65
12	北京航空航天大学 1 队	北京航空航天大学	张明远、伍俊洁、林家桢、潘叙辰	100.000	39.862	87
13	Cat	南京理工大学	赵金鑫、王芳、朱俊信、厉泉宏	100.000	39.029	70
14	HITSZ-MIPSbuilder	哈尔滨工业大学（深圳）	曹清宇、麦湘健	100.000	38.982	95
15	沙坪坝技校一队	重庆大学	葛胡军、柏蓉、刘亮、张启航	100.000	35.969	70
16	别急稳住我们能赢	华中科技大学	文宇鸿、付内东、吴敏康、李可欣	100.000	32.588	60
17	南航 1 队	南京航空航天大学	余浩岳、宗华、方梓威、秦毓泽	100.000	32.113	94
18	NKU 日新队	南开大学	林志翔、刘义情、卢宇航、李浩宇	100.000	30.589	85
19	西工大 2 队	西北工业大学	井幸斌、刘梦缘、刘艺杰	100.000	28.91	50
20	天津大学 2 队	天津大学	冯桥、王健宇、马泽雄、黄敬	100.000	26.35	60
21	NKU 月异队	南开大学	朱彦谕、刘炼、王继茗、郭佣鑫	100.000	24.539	110

希望以上 21 支队伍能够继续努力，在决赛中一展所长，都能取得优异的成绩。决赛将于 8 月 19-21 日在青海大学举行。

后续的话我还是觉得有必要实现一下 TLB（决赛要求），尽可能跑一下操作系统，虽然没进决赛，但还是要过一把决赛的瘾。

别的不多说，有收获就好（自我安慰），这个比赛也确实能让人学到了很多，从最开始 vivado 都不会装到后来能写出一个 MIPS CPU，从 verilog 只能写个全加器到后来写个 cache 模块，这中间的过程也着实学到了很多。最为简单的例子就是，以前很好奇执行一句 C 语言的 c = a + b，到硬件底层究竟是如何实现的，现在也确实能大概把他给讲清楚，同时也算是让我尝试了解了一下体系结构这个方向，以后会不会走这个方向也不好说。

简要记录，以此自勉。

2019.8.16 王赞