1 预赛提交

1.1 提交作品格式

预赛提交的作品需包含：预赛分数报告、设计文档、各测试项生成的 bit 流文件、基于 sram 或 axi 的 mycpu 源码。

提交目录参考预赛发布包（***nscscc2022_group_qualifier_submission***）里的 ***submission/MOU_1_zhangsan***，有以下注意事项：

(1) 不要提交整个预赛发布包，只需要将 ***submission/MOU_1_zhangsan*** 按格式整理好后打包提交即可。

(2) 请将 ***MOU_1_zhangsan*** 文件夹按“**学校英文简写_队伍编号_队长名拼音**”的格式进行**更名**。

(3) 整理设计文档：

将 CPU 设计文档整理为 pdf 格式，复制到 ***submission/MOU_1_zhangsan/***目录里。

(4) 填写功能和性能得分：

将功能和性能得分填写到 ***submission/MOU_1_zhangsan/score.xls*** 文件里。

(5) 整理预赛作品最后的 bit 流文件：

有 3 个 bit 流文件：功能测试 bit、记忆游戏 bit 和系统测试 bit，请分别放在 ***submission/MOU_1_zhangsan/bit/***目录里的对应子目录下。

请确保提交的 bit 流文件没有错误。

如果记忆游戏或系统测试未通过，则不需要提交对应的 bit 流文件。

(6) 整理预赛作品的 CPU 设计源码：

完成 SRAM 接口 CPU 的同学，请将 CPU 设计源码放在 ***submission/MOU_1_zhangsan/sram_src/mycpu/***目录中。直接做 AXI 接口 CPU 的同学，***sram_src*** 目录保持不变即可。

完成 AXI 接口 CPU 的同学，请将 CPU 设计源码放在 ***submission/MOU_1_zhangsan/src/mycpu/***目录中，同时将性能测试的 clk_pll IP 定制文件（***soc_axi_perf/rtl/xilinx_ip/clk_pll/clk_pll.xci***）复制到 ***submission/MOU_1_zhangsan/src/perf_clk_pll.xci***。

(7) 千万注意，CPU 设计源码中如果调用了 Xilinx IP（比如调用 Xilinx Block RAM IP），请将这些 IP 的定制文件*.xci 同样复制到 *submission/MOU_1_zhangsan/(sram_)src/mycpu/*目录中。也就是提交的 mycpu 源码应当包含除了大赛发布包提供的 SoC 相关文件外，你们所有新增的设计文件。

(8) ***submission/MOU_1_zhangsan/src*** 用于提交 AXI 接口的 mycpu 源码，***submission/MOU_1_zhangsan/sram_src*** 用于提交 SRAM 接口的 mycpu 源码，通常它们不会同时提交，只有在 soc_sram_func 通过 89 个功能点测试后，且尝试了 AXI 接口的 myCPU，才需要同时提交这两个目录。

预赛提交的作品的目录格式如下（***submission/MOU_1_zhangsan***）：

|-score.xls	Excel 表格，包含功能测试、性能测试得分的计算
|-**design.pdf**	PDF 文件，为 myCPU 设计报告
|--sram_src/	目录，SRAM 目录（可选）

| |--mycpu	目录，存放 SRAM 工程**新增的源码**文件，如果调用了 Xilinx IP，本目录下只提交 IP 的 xci 文件。注意不要忘记自行调用的 Xilinx IP 的 xci 文件。
|--src/	目录，AXI 目录
| |--mycpu	目录，存放 AXI 相关工程**新增的源码**文件，如果调用了 Xilinx IP，本目录下只提交 IP 的 xci 文件。注意不要忘记自行调用的 Xilinx IP 的 xci 文件。
| |--perf_clk_pll.xci	复制性能测试的 clk_pll.xci，更名为 perf_clk_pll.xci。不要忘记这一步。

1.2 提交作品确认

在预赛发布包（***nscscc2022_group_qualifier_submission***）中包含了 ***script*** 目录，该目录是我们的复核提交作品的半自动化测试环境。将其一起发布给大家，是希望参赛队伍提前使用该脚本确认自己提交的作品可以顺利生成各 bit 流文件。

因此，我们**强烈建议**在整理好提交作品后，运行一下 ***script*** 目录里的脚本，确认提交的作品没有问题。具体使用方法如下（也可以参考 ***script/Readme.txt***）：

(1) 确保 ***nscscc2022_group_qualifier_submission/submission*** 目录下有你的提交作品。

(2) 启动终端进入 ***nscscc2022_group_qualifier_submission/script*** 目录。

Linux 环境或 Windows 的 WSL 环境很好实现该步。

Windows 环境下也可以启动 CMD 终端，使用 cd 命令切换到 ***nscscc2022_group_qualifier_submission/script*** 目录。

(3) 在终端中输入并执行命令“***vivado -mode batch -source script.tcl***”进行 Vivado 的批处理模式。

在 Linux 环境或 Windows 的 WSL 环境，如果 Vivado 安装目录已添加到环境变量中，则直接输入以上命令即可。

在 Windows 环境下，我们也可以找到 Vivado 的执行脚本，比如可能是“***C:\Xilinx\Vivado\2019.2\bin\vivado.bat***”，输入命令“***C:\Xilinx\Vivado\2019.2\bin\vivado -mode batch -source script.tcl***”也可以进入 Vivado 的批处理模式。

(4) 接下来，我们可以输入“***help***”命令查看所有支持的命令。

目前支持命令有：help、list、init、runall、run、sim、download、exit。

(5) 首先输入命令“init all”完成你在 ***submission/***目录里放置的提交作品的工程初始化。

工程初始化会在 ***script/***目录里新建一个 ***project/***子目录，并会在 ***project/***目录中再新建一个你的作品的子目录，进而新建基于你的作品的功能测试、记忆游戏、性能测试和系统测试的 Vivado 工程。

(6) 然后输入 runall 命令执行 ***project/***子目录里的各 Vivado 工程的综合并生成 bit 流文件。

runall 命令会依次生成功能测试、记忆游戏、性能测试和系统测试的 bit 流文件。

如果 runall 有错，请在 ***project/***子目录里打开对应的 vivado 工程，查看错误原因，确认提交的源码是否符合规范，修改后重新执行(5)、(6)步。

如果 runall 顺利执行，则会在 ***result/***子目录里生成对应的 bit 流文件。

(7) 请将 ***result/***子目录里的各 bit 流文件依次下载到实验箱上，确认这些 bit 流文件可以正确运行。如果运行结果不符合预期，请检查提交的作品（脚本工程源码拷贝自 ***submission/MOU_1_zhangsan/src/***），修改后重新执行(5)、(6)、(7)步。

完成以上测试后，就可以放心的提交作品了。如果大家提交的作品都经过了以上测试，则我们后期使用 script 目录对所有作品进行批处理基本不会出错，会极大的方便我们的复核。

因此，在这里，希望大家提前测试一下自己的提交作品，感谢你们的配合。

1.3 提交方式

整个提交的压缩包应当不大于 100M，提交方式：

（1） 请将 ***MOU_1_zhangsan*** 文件夹按“**学校英文简写_队伍编号_队长名拼音**”的格式进行**更名**，并压缩，压缩包格式为 ZIP 格式，压缩之后的名称应与文件夹名称一致，如“MOU_1_zhangsan.zip”，不能包含中文。（如果一个学校只有一个队伍，则自动编号为 1；如果有多个队伍，不知道编号，请相互协商）。

（2） 压缩包请直接以邮件附件形式发送到 ***service@nscscc.com***，邮件名为：【2022 预赛作品提交】【xx 大学】【x 队】【队长名】【日期】，如【2022 预赛作品提交】【某大学】【1 队】【张三】【20220805】。

1.4 提交截止时间

预赛提交截止时间：2022 年 8 月 5 日 23:59:59。逾期不接收提交。

1.5 提交注意事项

在提交作品时，请确确认以下 10 项：

(1) 提交的目录结构符合规范；

(2) 确认包含 scode.xls 和 design.pdf 两个文档，注意命名；

(3) 确认包含功能测试目录 sram_src 或 src，只有指定情况才需要同时提交 sram_src 和 src；

(4) 如果有性能测试分，确认包含 src/ perf_clk_pll.xci；

(5) 如果 mycpu 调用了 xilinx IP，请确认这些 IP 的*.xci 文件已被赋值到(sram_)src/mycpu/目录中；

(6) 确认(sram_)src/mycpu 目录下，各 IP 只包含*.xci 文件。

(7) 确认 src/perf_clk_pll.xci 中的 PLL 出来的 sys_clk 未被修改，必须为 100MHz，cpu_clk 为自己性能测试的设定值；

(8) 确认性能测试环境里 src/perf_clk_pll.xci 中 cpu_clk 的设置满足时序要求，Implementation 后 WNS **非负值**；

(9) 提交的功能测试环境跑出的功能测试分与提交的功能测试分一致；

(10) 提交的性能测试环境跑出的功能测试分与提交的性能测试分一致。

提交格式或邮件格式不符合规范的，我们会酌情扣分，严重不符合记为 0 分。设计有所参考其他资料，但未在设计报告里指明引用关系的，我们会酌情扣分，严重者记为 0 分

包含但不只包含以下情况，功能测试复核分数一定会记为 0 分：

(1) 提交的代码添加到大赛发布包里的 SoC 里，无法运行仿真，或无法进行综合实现；

(2) 提交的代码添加到大赛发布包里的 SoC 里，生成的 bit 流与提交的 bit 流文件，运行结果不一致，且差异很大；

(3) 复核的 bit 流运行结果与提交的分数报告不一致，且差异很大；

(4) 提交的压缩包格式、邮件格式严重不符合要求的。

包含但不只包含以下情况，性能测试复核分数一定会记为 0 分：

(1) 提交的代码添加到大赛发布包里的 SoC 里，无法运行仿真，或无法进行综合实现；

(2) 提交的代码添加到大赛发布包里的 SoC 里，生成的 bit 流与提交的 bit 流文件，运行结果不一致，且差异较大；

(3) 复核的 bit 流运行结果与提交的分数报告不一致，且差异较大；

(4) 在 score.xls 中，计算性能分时，未通过的性能测试程序，其计时结果未按要求填写的；

(5) myCPU 使用的时钟非直接使用自 PLL 生成的 cpu_clk；

(6) PLL 的 sys_clk 擅自修改为非 100MHz；

(7) 复核时工程综合实现后 Implementation 栏的 WNS 为负值；

(8) 提交的压缩包格式、邮件格式严重不符合要求的；

1.6 预赛分数报告

预赛分数报告，也就是提交目录下的 ***MOU_1_zhangsan/score.xls***，应当为 Excel 文档，格式参考本目录下的 score.xls。

1.7 预赛设计文档

预赛设计文档，也就是提交目录下的 ***MOU_1_zhangsan/design.pdf***，应当为 PDF 文档。

1.8 设计注意事项

再次提醒参加大赛的各位同学，实现 CPU 时注意以下事项：

(1) 地址段 kseg1 是 Unmapped/Uncached，kseg0 是 Unmapped/可配置 Cached 属性的，它们都对应同一物理地址段。

(2) 对于 Uncached Load，强烈推荐将其实现为从原始存储区读回来：不能对 Uncached Load 进行 Load/Store 的前递，Load 返回的数据必须直接来自最终目的存储区。

(3) 强烈推荐对 Uncached Load 和 Uncached Store 保持顺序性：即使地址不同，Uncached Load 也不应该越过其前的 Uncached Store 先执行完成。

(4) AXI 协议不保证未完成的读、写请求的顺序性：先对地址 A 进行写，在该写未完成前，又发出对地址 A 的读，那么该读的返回数据可能是新值也可能是旧值，这没有违背 AXI 协议。