

# 1 预赛提交

## 1.1 提交目录格式

预赛提交的作品需包含：预赛分数报告、设计报告、功能测试环境 soc\_sram\_func 或 soc\_axi\_func、性能测试环境 soc\_axi\_func（如果无性能测试结果，则不提交该目录）和软件程序 Soft 目录。

提交目录参考大赛发布包 *Qualifiers\_Submission*，说明如下，注意：vivado 工程目录下只允许包含 xpr 和 bit 文件（自己生成的 bit 文件，请拷贝至该目录），其他目录和文件请删除；rtl/xilinx\_ip 目录下各 IP 目录里只包含 \*.xci 文件，其他目录和文件请删除；一般 soc\_axi\_func 和 soc\_sram\_func 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 soc\_axi\_func 和 soc\_sram\_func。

-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
-soc_axi_func/	目录，自实现 CPU 的功能测试环境：如果未实现 AXI 接口，则该目录为 soc_sram_func；如果 SRAM 接口 myCPU 通过 89 个功能点测试，且尝试了 AXI 接口，则需要同时提交 soc_sram_func 和 soc_axi_func。
--rtl/	目录，SoC_lite 的源码
--soc_lite_top.v	SoC_lite 的顶层
--myCPU /	目录，自实现 CPU 源码
--CONFREG/	目录，confreg 模块，连接 CPU 与开发板上数码管、拨码开关等 GPIO 类设备
--BRIDGE/	目录，bridge_1x2 模块，CPU 的 data sram 接口分流去往 confreg 和 data_ram
--xilinx_ip/	目录，Xilinx IP，包含 clk_pll、inst_ram、data_ram，只保留 *.xci 文件
--testbench/	目录，仿真文件
--mycpu_tb.v	仿真顶层，该模块会抓取 debug 信息与 trace_ref.txt 进行比对
--run_vivado/	目录，运行 Vivado 工程
--soc_lite.xdc	Vivado 工程设计的约束文件
--mycpu_prj1/	目录，Vivado2018.1 创建的 Vivado 工程 1
--mycpu.xpr	Vivado2018.1 创建的 Vivado 工程，可直接打开并进行仿真、综合实现且无错
--func.bit	Vivado2018.1 综合生成的 bit 文件，用于运行 89 个功能点测试
--memory.bit	Vivado2018.1 综合生成的 bit 文件，用于运行记忆游戏，没有通过记忆游戏测试则不包含。
-soc_axi_perf/	目录，自实现 CPU 的性能测试环境，如果无性能测试结果，则不提交该目录
--rtl/	目录，SoC_lite 的源码
--soc_lite_top.v	SoC_lite 的顶层
--myCPU /	目录，自实现 CPU 源码。应当与 soc_axi_func 里的 myCPU 完全一致
--CONFREG/	目录，confreg 模块，连接 CPU 与开发板上数码管、拨码开关等 GPIO 类设备
--ram_wrap/	目录，axi ram 的封装层，增加固定延迟设置
--xilinx_ip/	目录，Xilinx IP，包含 clk_pll、inst_ram、data_ram，只保留 *.xci 文件
--testbench/	目录，仿真文件
--mycpu_tb.v	仿真顶层，该模块会抓取 debug 信息与 trace_ref.txt 进行比对
--run_vivado/	目录，运行 Vivado 工程

		--soc_lite.xdc	Vivado 工程设计的约束文件	
		--mycpu_prj1/	目录, Vivado2018.1 创建的 Vivado 工程 1	
		--run_allbench.tcl	仿真依次运行 10 个性能测试程序的脚本	
			--mycpu.xpr	Vivado2018.1 创建的 Vivado 工程, 可直接打开并进行仿真、综合实现且无错
			--perf.bit	Vivado2018.1 综合生成的 bit 文件, 用于运行性能测试。
	-soft/			目录, 功能测试和性能测试软件程序目录, 应当是直接来自大赛发布包提供的文件。
		--func/		目录, 89 个功能点测试程序
		--memory_game/		目录, 记忆游戏测试程序
		--perf_func/		目录, 性能测试程序

## 1.2 提交方式

整个提交的压缩包应当不大于 100M, 提交方式:

- (1) 请将文件打包, 压缩包格式为 ZIP 格式, 命名为“**学校英文简写\_队伍编号\_队长名拼音.zip**”, 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; 如果有两个队伍, 不知道编号, 请相互协商分为 1、2 队)。
- (2) 压缩包请直接以邮件附件形式发送到 **service@nscscc.org**, 邮件名为: **【2018 预赛作品提交】【xx 大学】【x 队】【队长名】【日期】**, 如**【2018 预赛作品提交】【某大学】【1 队】【张三】【20180821】**。

## 1.3 提交截止时间

预赛提交截止时间: 2018 年 8 月 20 日 23:59:59。逾期不接受提交。

## 1.4 提交注意事项

在提交功能测试分和环境时, 请确认以下 13 项:

- (1) 提交的目录结构符合规范;
- (2) 确认包含 scode.doc 和 design.pdf 两个文档, 注意命名;
- (3) 确认包含功能测试目录 soc\_sram\_func 或 soc\_axi\_func, 只有指定情况才需要同时提交 soc\_sram\_func 和 soc\_axi\_func;
- (4) 如果有性能测试分, 确认包含 soc\_axi\_perf;
- (5) 确认性能测试目录 soc\_axi\_perf 里的 myCPU 和功能测试目录里的 myCPU 完全一致;
- (6) 自实现 CPU 的代码位于 rtl/myCPU 目录下, 且 vivado 工程是索引自此处的代码;
- (7) 除 rtl/myCPU/ 目录外, rtl/ 其余目录均没有被修改过;
- (8) 确认 rtl/Xilinx\_ip 目录下, 各 IP 目录里只包含\*.xci 文件。
- (9) 确认提交的 vivado 工程目录如要求一样干净, 且 xpr 和 bit 文件都是 Vivado2018.1 生成的, 且可直接打开且无错的运行仿真和性能测试。
- (10) 确认 myCPU 内部使用的时钟直接来自 PLL 出来的 cpu\_clk, 不允许进行内部分频;

- 
- (11) 确认性能测试环境里 `cpu_clk` 的设置满足时序要求, Implementation 后 WNS 非负值。
  - (12) 确认 `soft` 目录和大赛发布包发布的 *Qualifiers\_Submission/soft* 完全一致。
  - (13) 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。
  - (14) 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, 功能测试分数一定会记为 0 分:

- (1) 提交的工程无法打开, 或无法运行仿真, 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, 无法运行仿真, 或无法进行综合实现;
- (3) 提交的工程生成的 bit 流与提交的 bit 流文件, 运行结果不一致, 且差异很大;
- (4) 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;
- (5) (3)、(4)中任一 bit 流运行结果与提交的分数报告不一致, 且差异很大。
- (6) (3)、(4)中任一工程运行仿真的结果与提交的分数报告不一致, 且差异很大。
- (7) 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, 性能测试分数一定会记为 0 分:

- (1) 提交的工程无法打开, 或无法运行仿真, 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, 无法运行仿真, 或无法进行综合实现;
- (3) 提交的工程生成的 bit 流与提交的 bit 流文件, 运行结果不一致, 且差异很大;
- (4) 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;
- (5) (3)、(4)中任一 bit 流运行结果与提交的分数报告不一致, 且差异较大。
- (6) (3)、(4)中任一工程运行仿真的结果与提交的分数报告不一致, 且差异较大。
- (7) 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。
- (8) 提交的压缩包格式、邮件格式严重不符合要求的。
- (9) 提交的工程文件非 Vivado2018.1 创建的。
- (10) myCPU 使用的时钟非直接使用自 PLL 生成的 `cpu_clk`。
- (11) (3)、(4)中任一工程综合实现后 Implementation 栏的 WNS 为负值。

## 1.5 预赛分数报告

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

## 1.6 预赛设计报告

预赛设计文档, 也就是提交目录下的 *Qualifiers\_Submission/design.pdf*, 应当为 PDF 文档, 格式参考大赛发布

## 1.7 提交的功能测试 SoC

预赛提交的功能测试 SoC，应当来自功能测试包的 `soc_sram_func` 或 `soc_axi_func`，且组织结构类似。注意 vivado 工程脚本 `xpr` 应当已添加好 `myCPU`。vivado 工程目录下应当包含用于运行 89 个功能点测试和记忆游戏测试的 bit 流文件，如果未通过记忆游戏测试，则不包含记忆游戏的 bit 流文件。

如果后续大赛发布包，对功能测试包有更新，则应当使用最新的功能测试包里的 `soc_sram_func` 或 `soc_axi_func`。

## 1.8 提交的性能测试 SoC

预赛提交的性能测试 SoC，应当来自性能测试包的 `soc_axi_perf`，且组织结构类似。注意 vivado 工程脚本 `xpr` 应当已添加好 `myCPU`。vivado 工程目录下应当包含用于运行联合编译后的性能测试程序的 bit 流文。

如果所有性能测试都未通过，则不提交该目录。

如果后续大赛发布包，对性能测试包有更新，则应当使用最新的性能测试包里的 `soc_axi_perf`。

## 1.9 提交的 soft 目录

预赛提交的 Soft 目录，应当来自本提交目录的 `soft` (*Qualifiers\_Submission/soft*)，且完全一致。*Qualifiers\_Submission/soft* 是将发布包里的功能测试和性能测试程序整合到了一个目录。

如果后续大赛发布包，对 *Qualifiers\_Submission/soft* 目录有更新，则应当使用最新的 *Qualifiers\_Submission/soft*。

## 1.10 设计注意事项

在此，提醒参加大赛的各位同学，实现 CPU 时注意以下事项：

- (1) CP0 寄存器 Status 实现 bev 位，恒为 1 即可。
- (2) 地址段 `kseg1` 是 Unmapped/Uncached，`kseg0` 是 Unmapped/可配置 Cached 属性的，它们都对对应同一物理地址段。
- (3) Confreg 模块分配的地址段是 `0xffff_0000~0xffff_ffff`，它们应该设置为 Uncached 属性，且不支持 burst 访问。
- (4) 对于 AXI 接口的 CPU，功能测试和性能测试的 `myCPU` 代码应当是完全一致的。