

# CPU31 网站提交说明

## 一、网站提交说明

1. 网站自动验证程序采用前仿真

2. 需配置 IP 核，IP 核配置参照“关于CPU测试.pdf”的9页到12页。

3. 每条指令都会测试，大家可以用文件“31条CPUtest指令示例和使用说明”中的每个测试程序对你们自己做的 CPU 分别进行测试，测试结果和 Mars 输出结果进行对比（以 Mars 为标准）。

4. 接口说明：

由于服务器测试程序是对寄存器的值等进行测试，所以需要统一一些接口和模块名：

a) sccomp\_dataflow 模块接口（即顶层模块）按照如下定义：

```
module sccomp_dataflow(  
    input clk_in,  
    input reset,  
    output [31:0] inst,  
    output [31:0] pc  
);
```

提交作业自动验证使用的是服务器中tb文件，验证时又要检查寄存器堆中寄存器的数据，所以需要作一定的约束，希望大家按照要求提交作业。在 sccomp\_dataflow 模块中实例化cpu 模块务必按照 cpu sccpu(……)命名定义，在 cpu 模块中实例化寄存器堆模块 regfile，需要按照 regfile cpu\_ref(……)命名进行定义，在 regfile 模块中定义 32 个寄存器需按照“reg [31:0] array\_reg[31:0];”命名进行定义。现把测试文件中部分输出写法给你们参考，以理解上述说明：

```
$fdisplay(file_output, "pc: %h", pc);  
$fdisplay(file_output, "instr: %h", inst);  
$fdisplay(file_output, "regfile0: %h", test.uut.sccpu.cpu_ref.array_reg[0]);  
$fdisplay(file_output, "regfile1: %h", test.uut.sccpu.cpu_ref.array_reg[1]);
```

（其中 sccpu、cpu\_ref、array\_reg 三个变量按照上述命名说明进行理解）

sccomp\_dataflow 作为顶层模块，主要包括 dmem、imem 和 cpu 模块的实例化，cpu31 的主要实现部分是在 cpu 模块中，包括寄存器堆的调用也是在该模块，大家可能写法稍微不同，不过为了统一测试，希望大家配合接口验证。

5. 网站验证需要提交的文件:

- a) 工程中所有自己编写的.V 文件
- b) 使用了 IPcore (imem), 故需要将相应的.V 文件和.mif 文件加入文件夹

注意: mif 需要提前初始化好, 即生成 ipcore 时使用 coe 文件赋值, 具体做法如下: 使用所给的 “mips\_31\_mars\_simulate.coe” 生成 ipcore (注意模块实例化名的替换), 然后在相应路径下 (默认是 “./cpu31.srcs/sources\_1/ip/imem”, ‘imem’ 是你们自己定义的名字) 找到和 ipcore 名字对应的.V 文件和.mif 文件, 以及自动生成的 “dist\_mem\_gen\_v8\_0.v” 加入到文件夹, 和自己编写的源文件都放在同一层目录打包成.zip格式提交。