

计算机设计与实践

基于Trace的CPU功能验证

2023·夏

哈工大



HITSZ 实验与创新实践教育中心
Education Center of Experiments and Innovations, HITSZ

实验目的

- ◆ 理解基于Trace的CPU功能验证方法的基本原理
- ◆ 掌握如何使用Trace验证方法



实验内容

- ◆ 为CPU添加Trace验证所需的调试信号
- ◆ 在虚拟机环境中运行Trace验证，并据此定位并修复错误



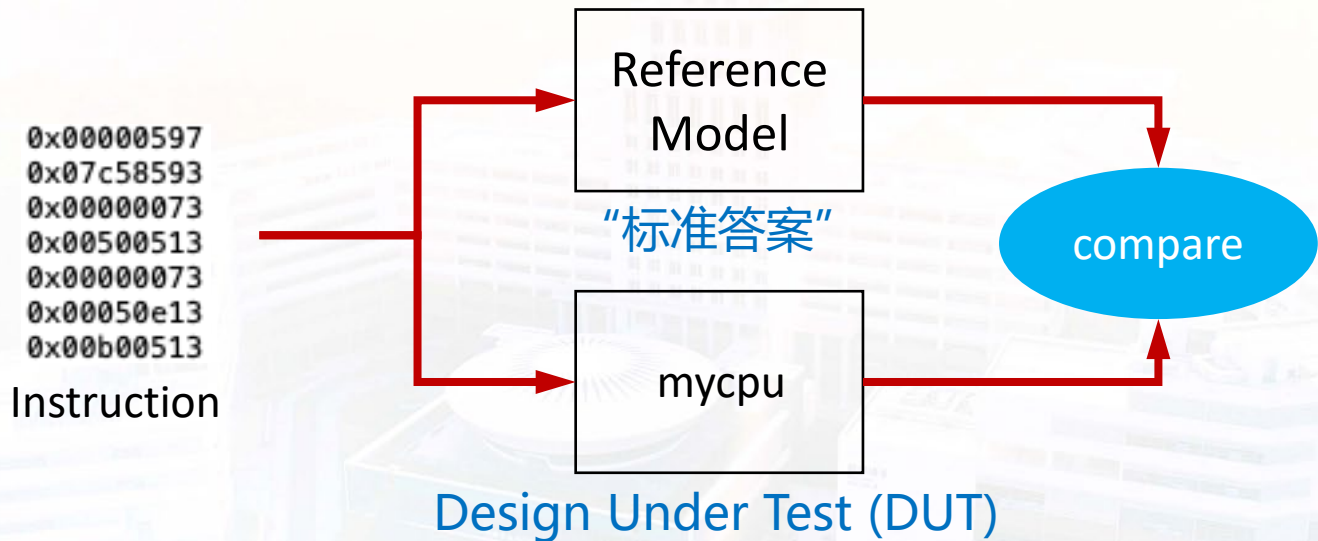
Trace比对

- ◆ Trace: CPU执行指令序列时产生的信息 (PC、**写寄存器**的信息, etc)
- ◆ 基于Trace比对的验证方法:
 - ① 用已知功能正确的CPU运行测试程序, 记录Trace0 (Golden Trace)
 - ② 用待验证CPU运行测试程序, 产生Trace1
 - ③ 将Trace1与Trace0进行实时比对, 如果出现不同, 立即报错并停止
- ◆ 特殊情况:
 - ◆ Store指令 —— 后续相关的Load指令写入寄存器的值与GT比对



Trace比对

◆ Trace比对的基本原理:



Trace比对

◆ Trace-检查写寄存器:

CPU每个周期完成1条指令，但并不是每个周期都有指令到达WB阶段，因此检查写寄存器要在WB阶段有指令时进行

```
output      debug_wb_have_inst, // 此时刻，WB阶段有指令（对于单周期CPU，恒为1）
output [31: 0] debug_wb_pc,      // WB阶段的PC（如果wb_have_inst=0，此项可为任意值）
output      debug_wb_ena,        // WB阶段寄存器写使能（如果wb_have_inst=0，此项可为任意值）
output [4: 0] debug_wb_reg,      // WB阶段写入的寄存器（如果wb_ena=0 | wb_have_inst = 0，此项可为任意值）
output [31:0] debug_wb_value     // WB阶段写入寄存器的值（如果wb_ena=0 | wb_have_inst = 0，此项可为任意值）
```

have_inst信号是表示CPU完成了一条指令，需要进行比对
测试系统发现此信号拉高后，将触发一次Trace比对



Trace比对 – 要求

- ◆ 跑Trace时，取消注释defines.vh中的RUN_TRACE宏定义

```
≡ defines.vh ×  
C: > Users > csmj > AppData > Local > Temp > 360zip$Temp > 36  
1 // Annotate this macro before synthesis  
2 //`define RUN_TRACE
```

- ◆ IROM/DRAM

模块名	IROM	DRAM
大小	32bit×65536	

模块名和大小都不要写错，否则trace不过!

- ◆ PC初始值: 0x0000_0000
 - ◆ 复位后CPU执行的首条指令的地址是0x0000_0000



Trace比对 – 使用

◆ RTL集成

➤ 顶层文件:

- ① 跑trace时取消注释头文件的RUN_TRACE宏定义
- ② IROM和DRAM只需实例化

```
1|`include "defines.vh" // 运行Trace测试时，将此文件的RUN_TRACE取
2|module miniRV_SoC (
3|    input wire      fpga_rst,           // High active
4|    input wire      fpga_clk,
5|    ..... // 外设I/O接口信号
6|`ifdef RUN_TRACE
7|    ,// Debug Interface
8|    output wire      debug_wb_have_inst, // WB阶段是否有指
9|    output wire [31:0] debug_wb_pc,      // WB阶段的PC (若
10|    output wire      debug_wb_ena,       // WB阶段的寄存器
11|    output wire [4:0] debug_wb_reg,      // WB阶段写入的寄
12|    output wire [31:0] debug_wb_value    // WB阶段写入寄存
13|`endif
14|);
15|    .....
16|
17|    myCPU Core_cpu (
18|        .cpu_rst    (fpga_rst),
19|        .cpu_clk     (cpu_clk),
20|        //.....
21|    );
22|
23|    // 下面两个模块，只需要实例化代码和连线代码，不需要创建IP核
24|    IROM Mem_IROM (
25|        .a          (...),
26|        .spo         (...),
27|    );
28|
29|    DRAM Mem_DRAM (
30|        .clk         (...),
31|        .a           (...),
32|        .spo         (...),
33|        .we          (...),
34|        .d           (...),
35|    );
36|
37|endmodule
```



Trace比对 – 使用

- ◆ Trace使用

- ◆ 环境：VirtualBox虚拟机、远程实验平台
- ◆ 使用：git clone下载Trace测试包，将SoC源码拷贝至mySoC目录，在cdp-tests目录输入命令make即可
- ◆ 详细说明及镜像下载：<https://hitsz-cslab.gitee.io/cpu/trace/>





HITSZ 实验与创新实践教育中心
Education Center of Experiments and Innovations, HITSZ