

# 计算机组成原理

## 实验3 高速缓存器设计

2024 · 春

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

---

- 掌握直接映射Cache的基本结构及工作原理
- 掌握直接映射Cache的HDL实现方法
- 认识Cache对计算机系统性能的影响



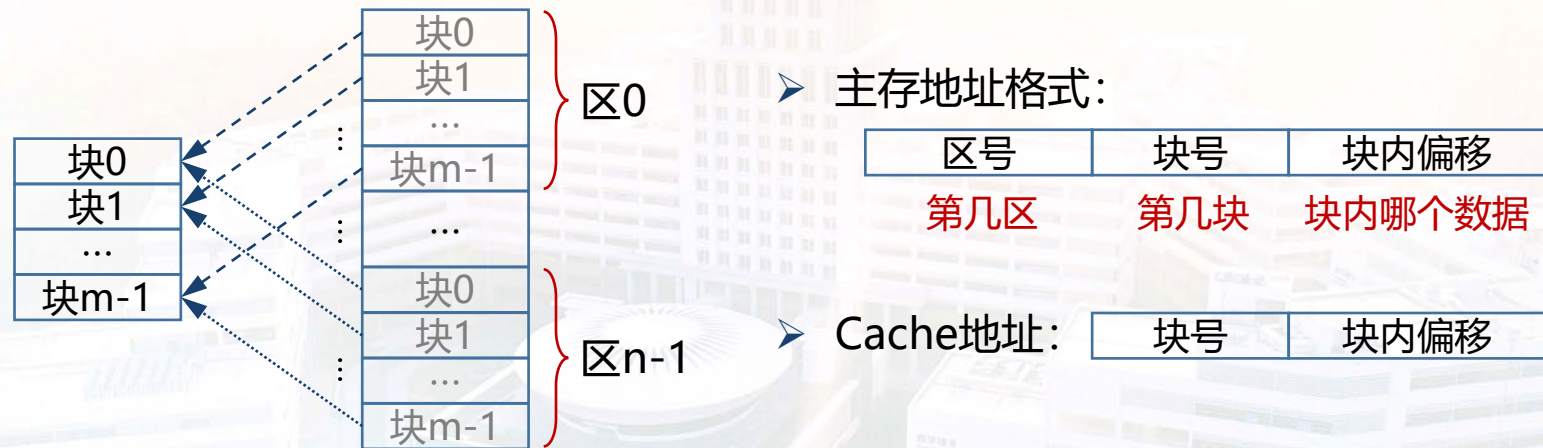
# 实验内容

- 在上一个实验的SoC工程中，设计实现直接映射 ICache：
  1. 完成SoC工程中的 ICache 模块 ([ICache.v](#))
  2. 主存大小为**32KB**，采用Block Memory存储介质（随机延迟）
  3. Cache大小为**1KB**，块大小为**16B**，采用Block Memory存储介质
  4. ICache对CPU是只读的，故需处理**读命中**、**读缺失**
  5. 测试并**比较**无ICache和有ICache时的测试**程序运行时间**，记录命中率

# 实验原理

## 1. 直接映射方式

- 主存被分为n个区，每个区的大小均与Cache相同



- 每个区中的第i块都能放在Cache中的第i块 —— 将区号作为tag来区分

# 实验原理

## 2. 地址分解

- 例：主存大小为256KB，Cache大小为4KB，数据块大小为128B

主存 **256KB** -> 主存地址**18bit**

Cache **4KB** -> Cache地址**12bit**

数据块 **128B** -> 块内偏移**7bit**

- 主存地址格式：



块号位宽 = **12bit** - **7bit** = **5bit**

区号位宽 = **18bit** - **12bit** = **6bit**

- Cache地址：



# 实验原理

## 3. Cache结构及访问原理

➤ Cache块由有效位、块标签、数据块组成

- 有效位 (v) : Cache块初始时无效,  
缓存内存数据后有效
- 块标签 (tag) : 主存地址的区号
- 数据块 (data) : 多个连续的数据单元

v	tag	数据块			
⋮	⋮	⋮			

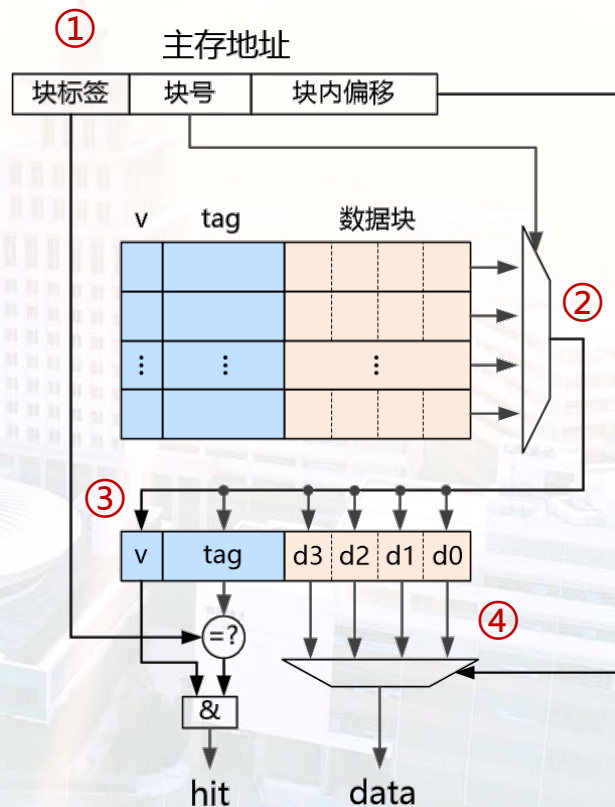


# 实验原理

## 3. Cache结构及访问原理

### ➤ Cache访问原理

- ① 地址划分  
分解主存地址
- ② Cache块寻址  
使用块号取出Cache块
- ③ 命中判断  
 $v \& (tag1 == tag2)$
- ④ 取出数据  
使用块内偏移选出数据块



# 实验原理

## 3. Cache结构及访问原理

### ➤ Cache访问原理

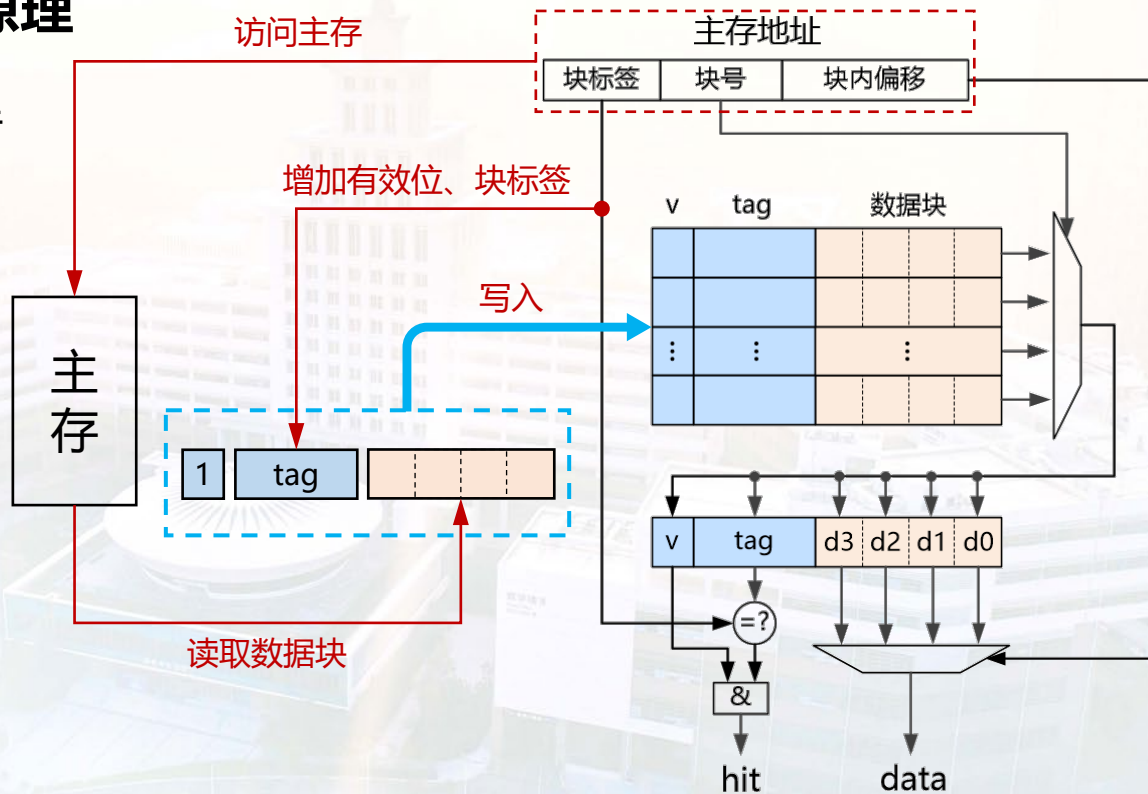
#### ⑤ 缺失处理

读取内存

增加信息

写入Cache

判断命中





# 实验原理

## 4. Cache的读状态机

- **IDLE**: 就绪状态, 可以接收读请求
- **TAG\_CHK**: 检查Cache是否命中
- **REFILL**: 读取主存, 为数据块增加有效位和块标签, 再将新的数据块写入Cache存储体



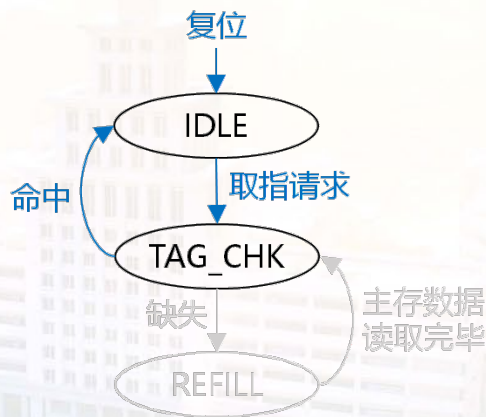
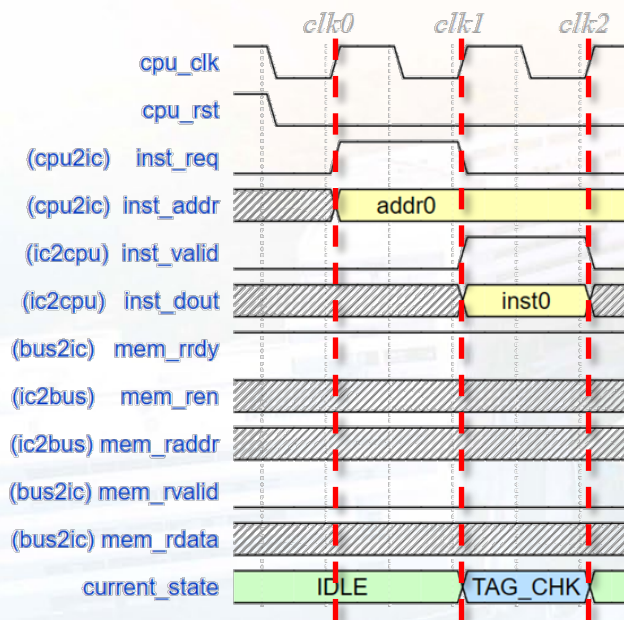
# ICache.v 接口信号

名称	位宽	属性	功能描述
cpu_clk	1	输入	时钟信号
cpu_rst	1	输入	复位信号（高电平有效）
与CPU的接口			
inst_rreq	1	输入	CPU的取指请求
inst_raddr	32	输入	CPU的取指地址
inst_valid	1	输出	指令有效信号（命中信号）
inst_out	32	输出	返回给CPU的指令
与总线模块的接口			
mem_rrdy	1	输入	总线就绪信号（有效时ICache才能发出读请求）
mem_ren	4	输出	读使能信号（对ICache，取值只能是4'h0或4'hF）
mem_raddr	32	输出	读主存的地址信号
mem_rvalid	1	输入	总线返回的主存数据有效信号
mem_rdata	128	输入	总线返回的主存数据

# 实验原理

## 5. Cache的读时序

### ① 读命中



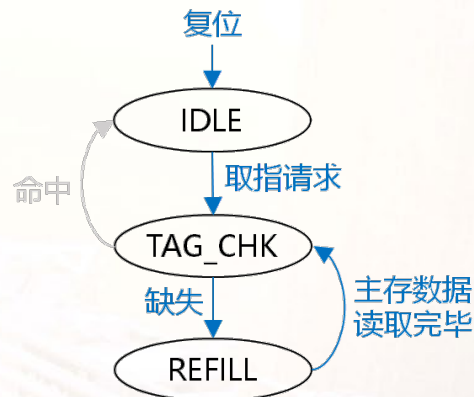
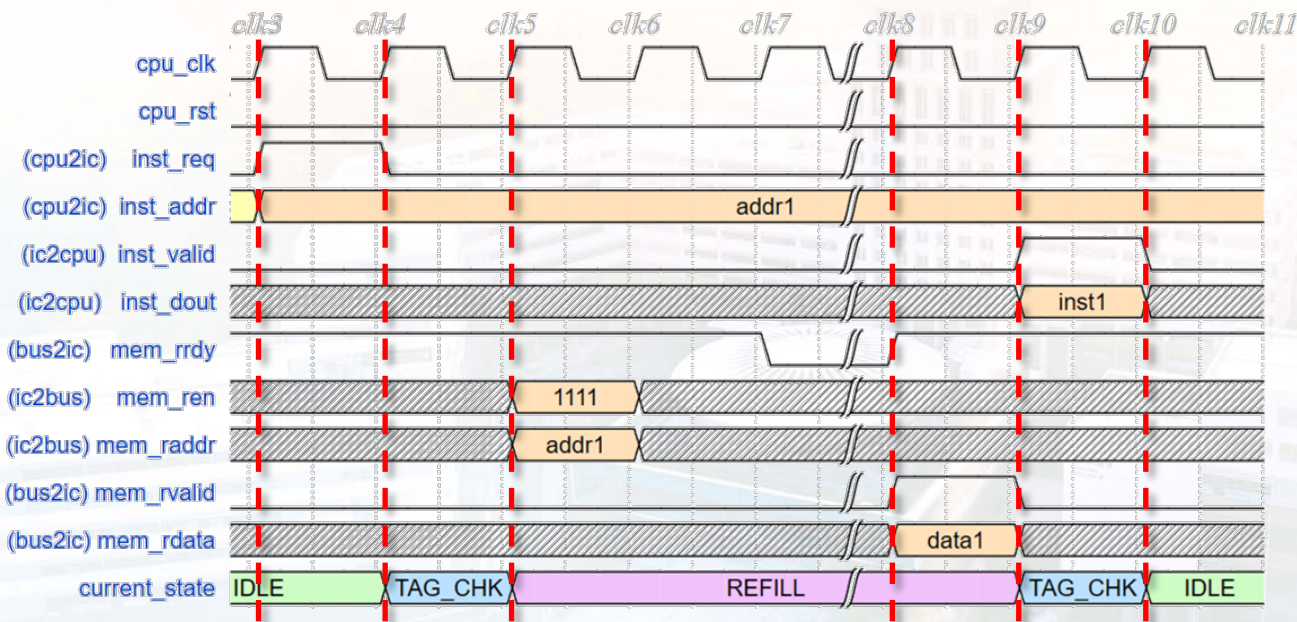
【clk0】CPU发出读请求

【clk1】Cache命中，输出命中信号和指令

# 实验原理

## 5. Cache的读时序

### ② 读缺失



【clk3】CPU发出读请求

【clk4】Cache缺失

【clk5】rrdy时读主存

【clk8】主存数据返回

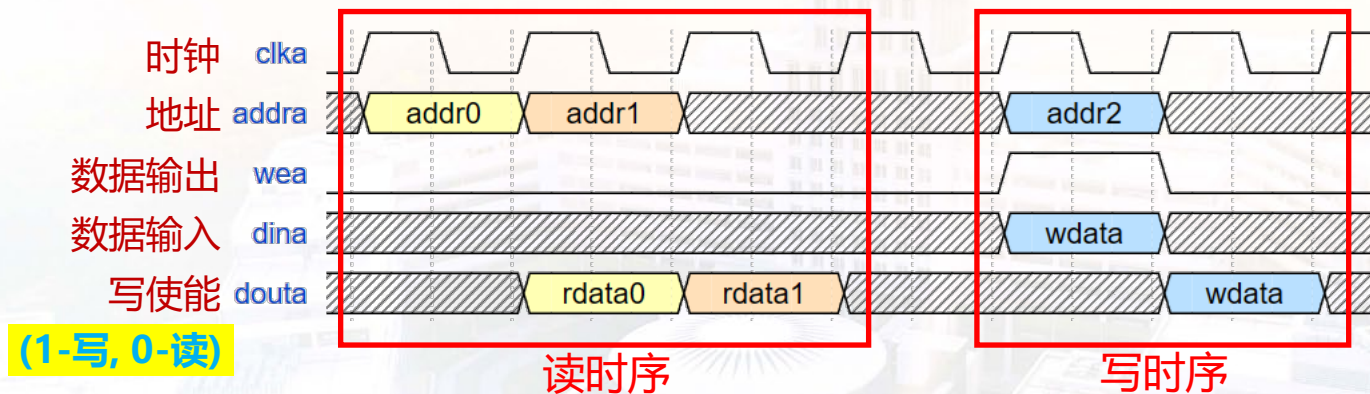
【clk9】命中并返回指令



# 实验原理

## 6. Cache存储体

- Cache采用Block Memory作为存储介质，已在模板代码中实例化好



- 读时序：当前时钟给出地址，下个时钟输出数据，可连续读取
- 写时序：当前时钟给出写地址、写数据、写使能wea，下个时钟数据被写入，且新写的数据被输出在douta端口上

# 实验步骤

1. 理解直接映射Cache的结构及工作原理
2. 打开上一个实验的SoC工程，将头文件defines.vh的前两行代码取消注释

defines.vh

```
1|// `define RANDOM_DELAY  
2|// `define ENABLE_ICACHE
```



defines.vh

```
1|`define RANDOM_DELAY  
2|`define ENABLE_ICACHE
```

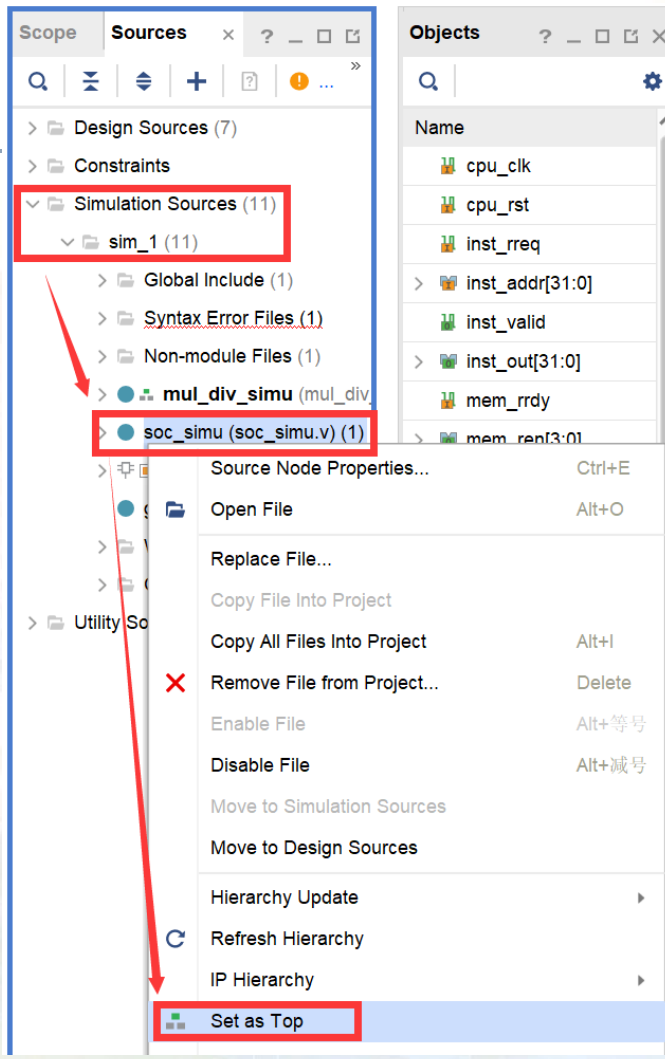
3. 按[实验指导书-实验步骤](#)，粘贴模板代码到ICache.v
4. 根据模板代码的提示，完成ICache.v，包括划分主存地址、编写命中判断的组合逻辑、设计状态机，etc.
5. 运行功能仿真，并根据输出的调试信息，结合仿真波形，定位错误点



# 实验步骤

5. 运行功能仿真，并根据输出的调试信息，结合仿真波形，定位错误点

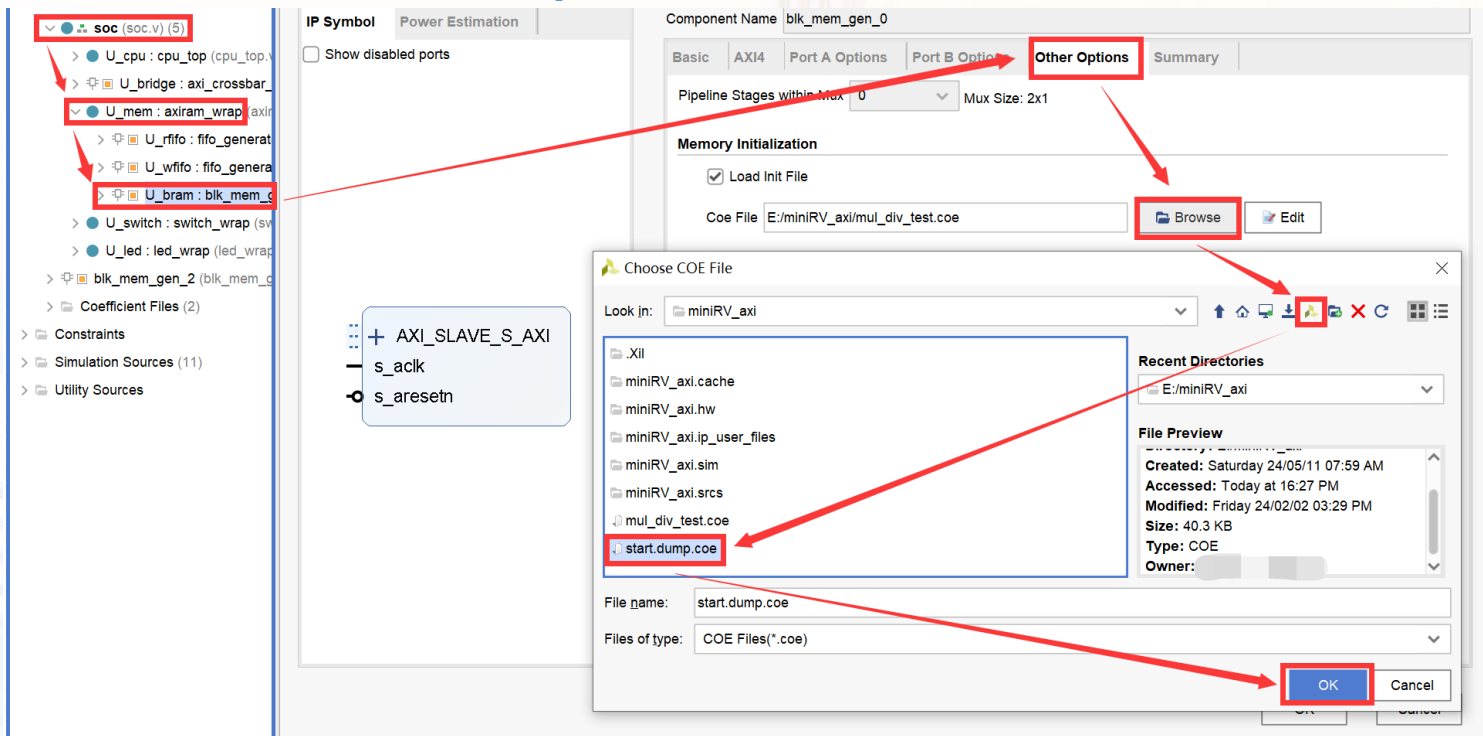
① 修改仿真顶层文件为 **soc\_simu.v**



# 实验步骤

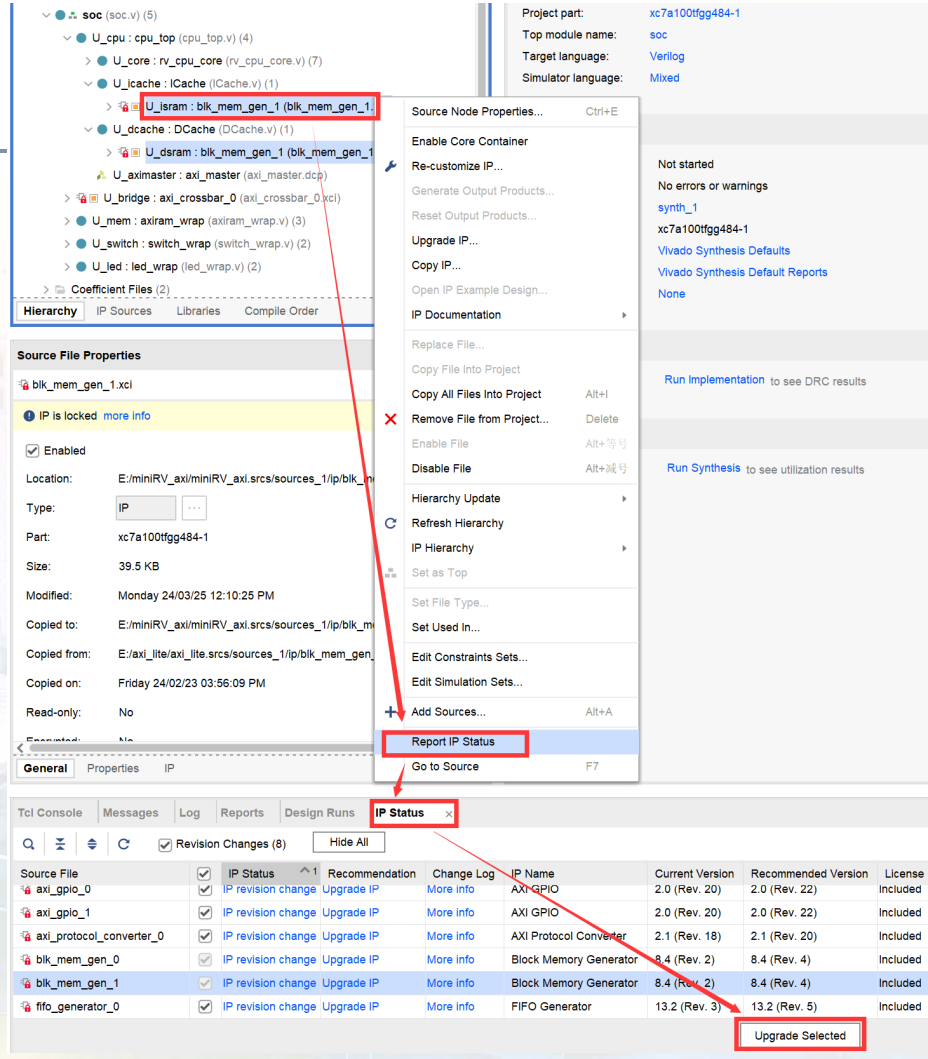
## 5. 运行功能仿真，并根据输出的调试信息，结合仿真波形，定位错误点

### ② 修改测试程序为start.dump.coe



5. 运行功能仿真，并根据输出的调试信息，结合仿真波形，定位错误点

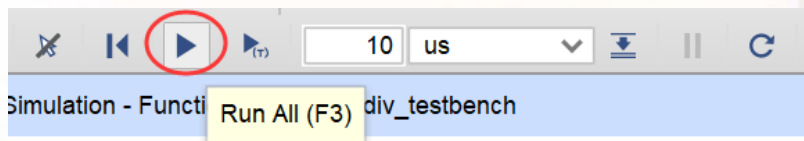
➤ 若使用2018.3以上的版本，  
需升级IP核



# 实验步骤

5. 运行功能仿真，并根据输出的调试信息，结合仿真波形，定位错误点

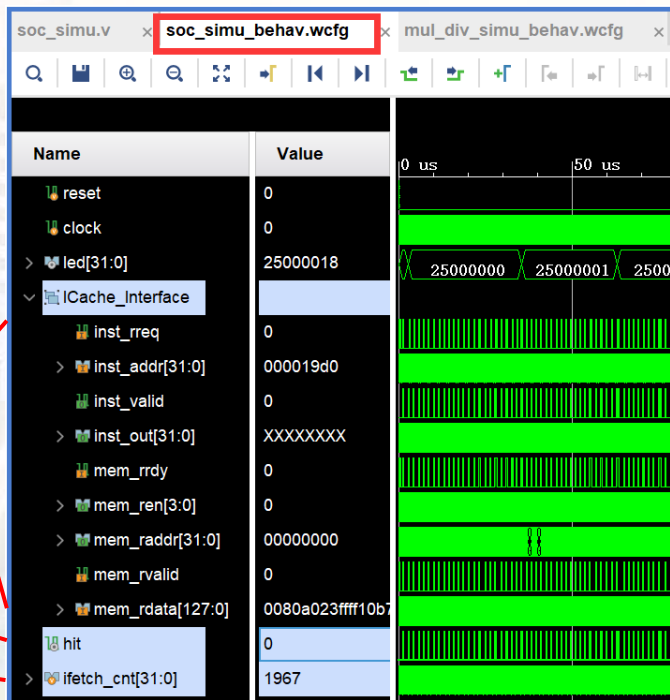
③ 点击“Run All”运行功能仿真，在soc\_simu\_behav.wcfg中查看波形



➤ 调试时，根据需要自行添加ICache.v内部的信号到波形中

ICache.v接口信号

ICache命中信号  
取指计数器

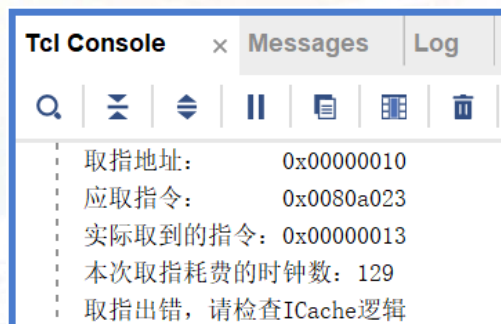


# 实验步骤

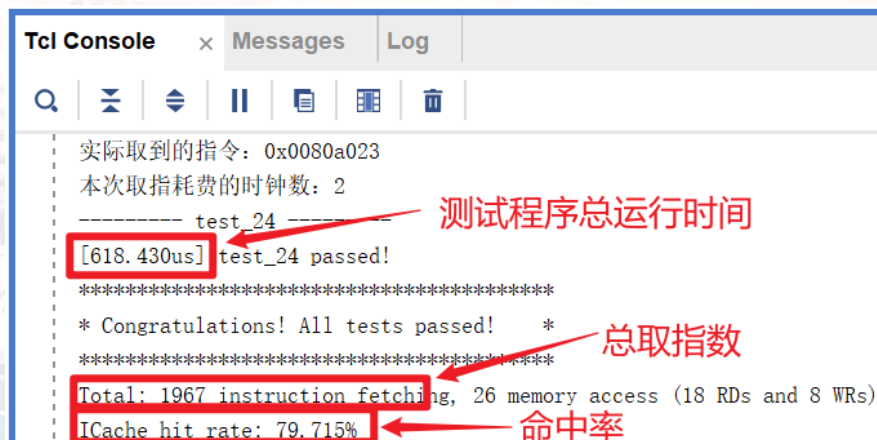
5. 运行功能仿真，并根据输出的调试信息，结合仿真波形，定位错误点

④ 结合控制台信息进行调试

## 测试失败



## 测试通过



# 实验步骤

## 6. 测试并对比有ICache和无ICache时，SoC运行测试程序的时间，并记录命中率

### ① 打开ICache，运行仿真，记录程序运行时间

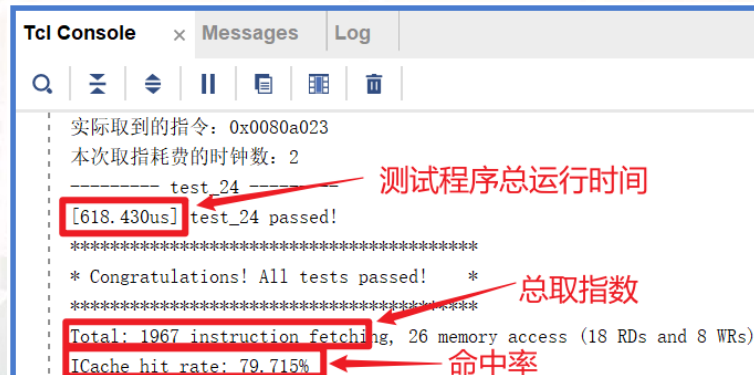
defines.vh

```
1|`define RANDOM_DELAY
2|`define ENABLE_ICACHE
```

### ② 关闭ICache，运行仿真，记录程序运行时间

defines.vh

```
1|`define RANDOM_DELAY
2|// `define ENABLE_ICACHE
```





# 验收与提交

- **验收内容**

- 课上检查使能ICache的SoC是否通过测试程序：**3分**

- **提交内容**

- ICache模块的.v源文件：**1分**
  - 实验报告（按模板完成）：**3分**
- 将上述文件打包成.zip，以“学号\_姓名.zip”命名提交到作业系统
  - ◆ 注意：**如有雷同，双方均0分！**



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