

计算机组成原理

实验3 高速缓存器设计

2025 · 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

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



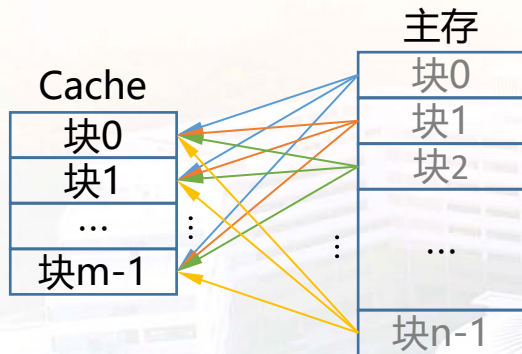
实验内容

- 在上一个实验的SoC工程中，设计实现全相联映射 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**128B**，块大小为**32B**，采用二维reg数组作为存储介质
 4. ICache对CPU是只读的，故只需处理**读命中**、**读缺失**
 5. 发生Cache块替换时，采用**随机替换**策略
 6. 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. 全相联映射方式

- 主存被分为n个块，每个块的大小均等于Cache数据块大小



- 主存地址格式:

主存块号	块内偏移
第几块	块内哪个数据

- Cache地址:

Cache块号	块内偏移
---------	------

- 主存数据块可放在Cache内任意一块 —— **用主存块号作为tag来区分**

实验原理

2. 地址分解

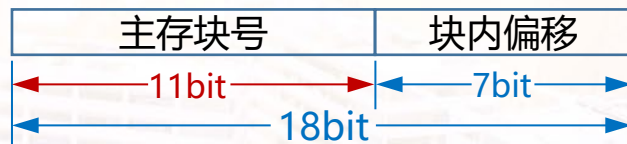
- 例：主存大小为256KB，Cache大小为4KB，数据块大小为128B

主存 **256KB** -> 主存地址**18bit**

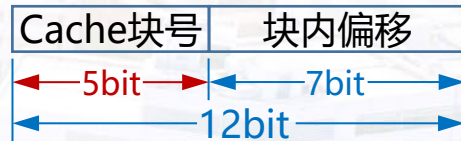
Cache **4KB** -> Cache地址**12bit**

数据块 **128B** -> 块内偏移**7bit**

- 主存地址格式：



- Cache地址：



主存块号位宽 = **18bit** - **7bit** = **11bit**

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

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,
缓存主存数据后有效
- 块标签 (tag) : 主存地址的块号
- 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

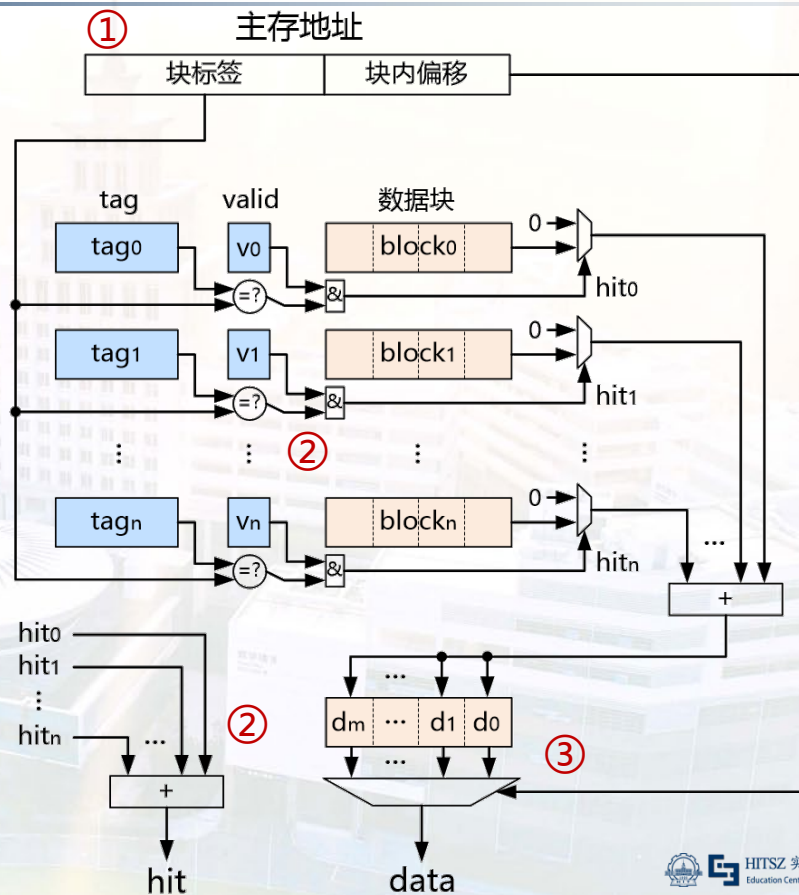
分解主存地址

② 命中判断

$(tag_{mem} == tag_i) \& v_i$

③ 取出数据

使用块内偏移选出数据



实验原理

3. Cache结构及访问原理

➤ Cache访问原理

④ 缺失处理

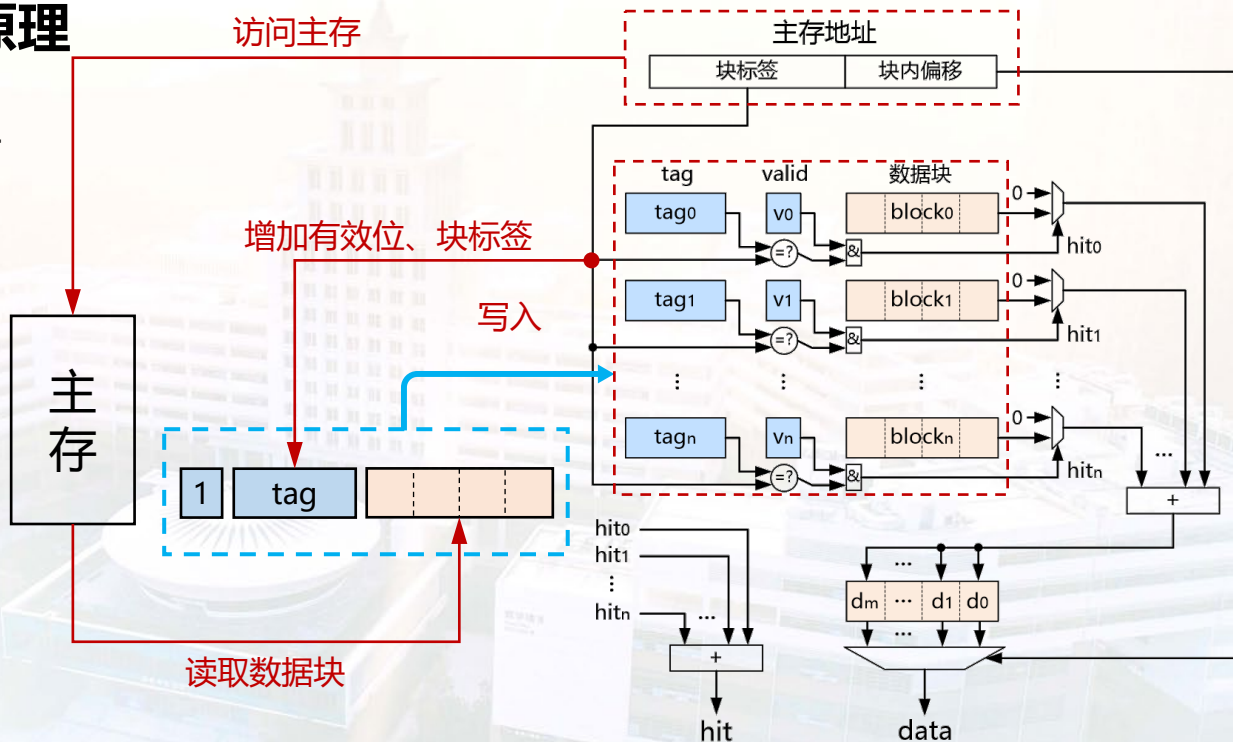
读取内存

增加tag、v

替换Cache块

再次判断命中

输出数据



实验原理

4. Cache块的装填&替换

- Cache采用reg型二维数组作为存储介质

```
1| reg [?:0] valid;           // 有效位
2| reg [?:0] tag   [3:0];     // 块标签
3| reg [?:0] data  [3:0];     // 数据块
```

- free_index指向空闲块, replace_index指向将要被替换的Cache块

	v	tag	数据块			
free_index →	0					
	0					
	⋮	⋮				
	0					

实验原理

4. Cache块的装填&替换

- Cache采用reg型二维数组作为存储介质

```
1| reg [?:0] valid;           // 有效位
2| reg [?:0] tag   [3:0];     // 块标签
3| reg [?:0] data  [3:0];     // 数据块
```

- free_index指向空闲块, replace_index指向将要被替换的Cache块

	v	tag	数据块			
	1					
free_index →	0					
	⋮	⋮				
	0					

实验原理

4. Cache块的装填&替换

- Cache采用reg型二维数组作为存储介质

```
1| reg [?:0] valid;           // 有效位
2| reg [?:0] tag [3:0];       // 块标签
3| reg [?:0] data [3:0];      // 数据块
```

- free_index指向空闲块, replace_index指向将要被替换的Cache块

v	tag	数据块			
1					
1					
⋮	⋮				
1					

free_index →

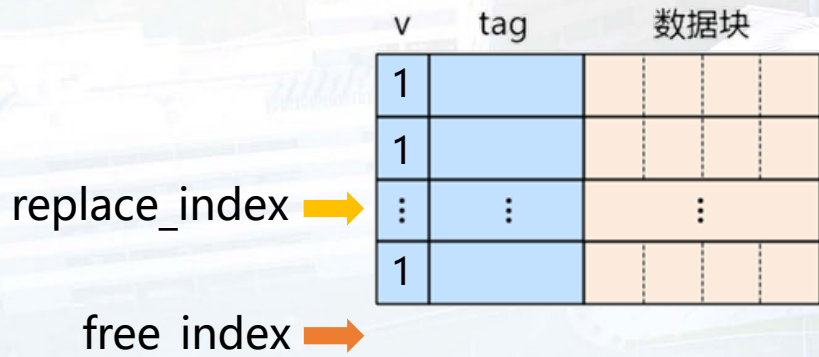
实验原理

4. Cache块的装填&替换

- Cache采用reg型二维数组作为存储介质

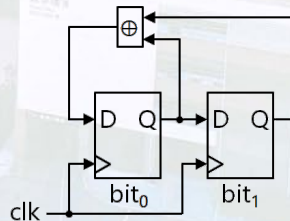
```
1| reg [?:0] valid;           // 有效位
2| reg [?:0] tag   [3:0];      // 块标签
3| reg [?:0] data  [3:0];      // 数据块
```

- free_index指向空闲块, replace_index指向将要被替换的Cache块



- 随机替换策略

- replace_index = LFSR随机数
- 4个Cache块, 需2bit随机数



实验原理

5. 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

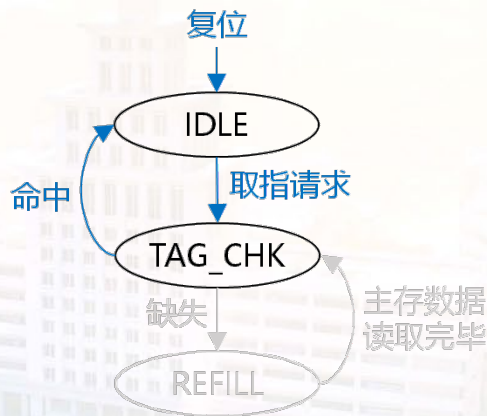
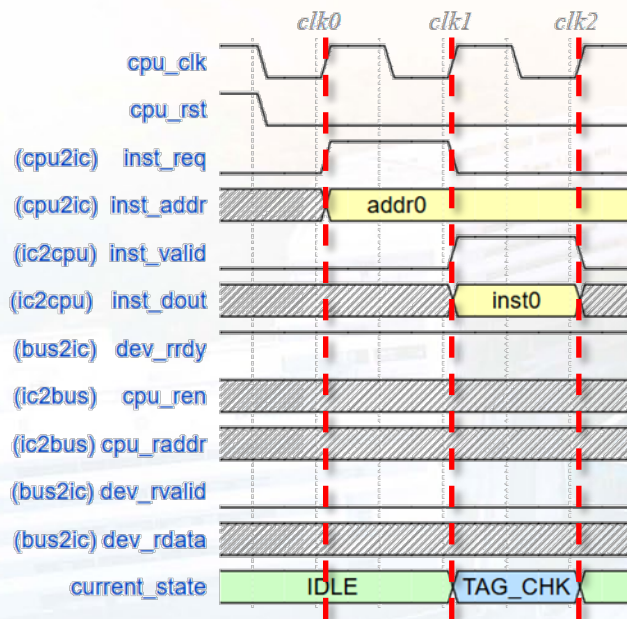
接口信号

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			
dev_rrdy	1	输入	总线就绪信号（有效时ICache才能发出读请求）
cpu_ren	4	输出	读使能信号（对ICache，取值只能是4'h0或4'hF）
cpu_raddr	32	输出	读主存的地址信号
dev_rvalid	1	输入	总线返回的主存数据有效信号
dev_rdata	256	输入	总线返回的主存数据

实验原理

6. Cache的读时序

① 读命中



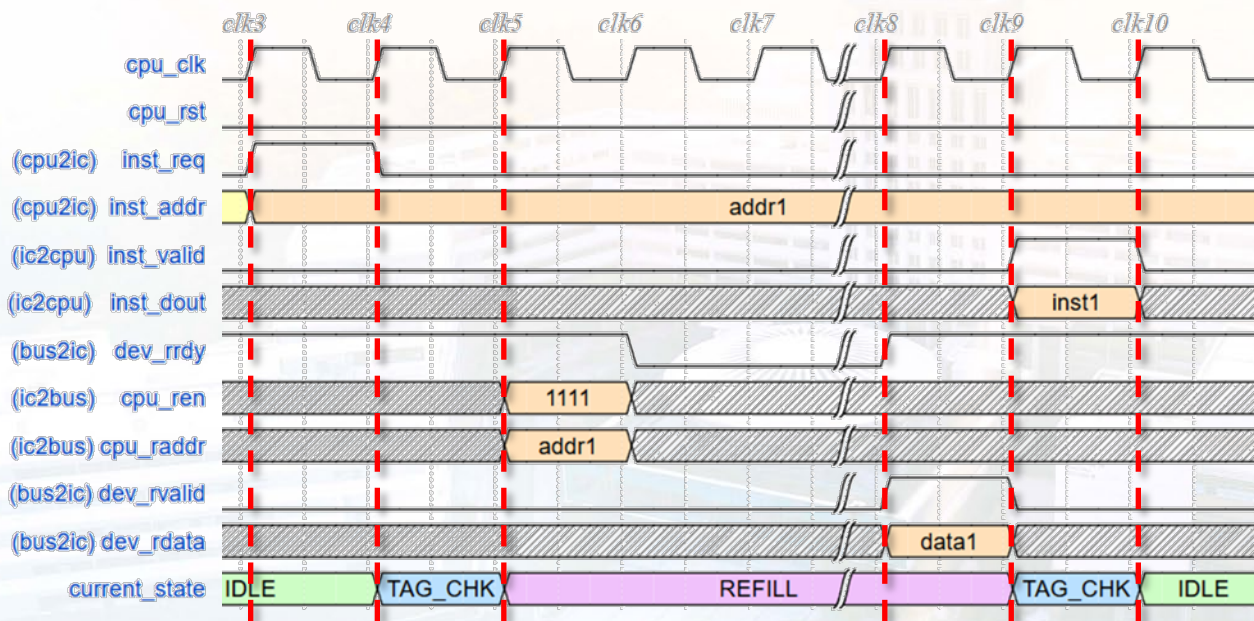
【clk0】CPU发出读请求

【clk1】Cache命中，输出命中信号和指令

实验原理

6. Cache的读时序

② 读缺失



【clk3】CPU发出读请求

【clk4】Cache缺失

【clk5】dev_rrdy时读主存

【clk8】主存数据返回

【clk9】命中并返回指令

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

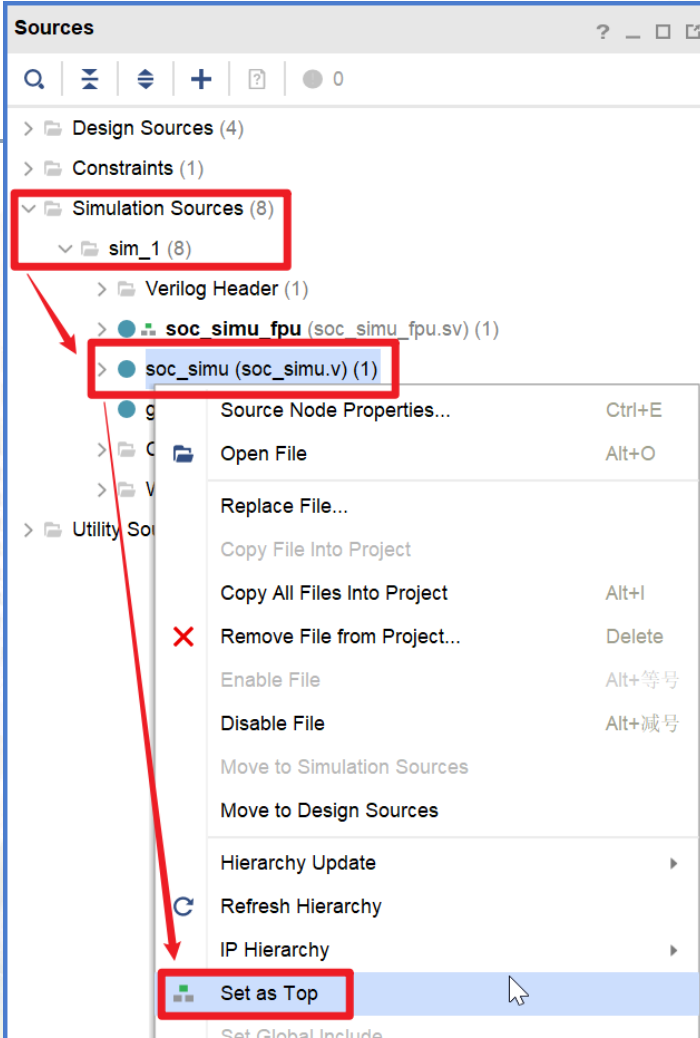
实验步骤

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**

soc_simu.v

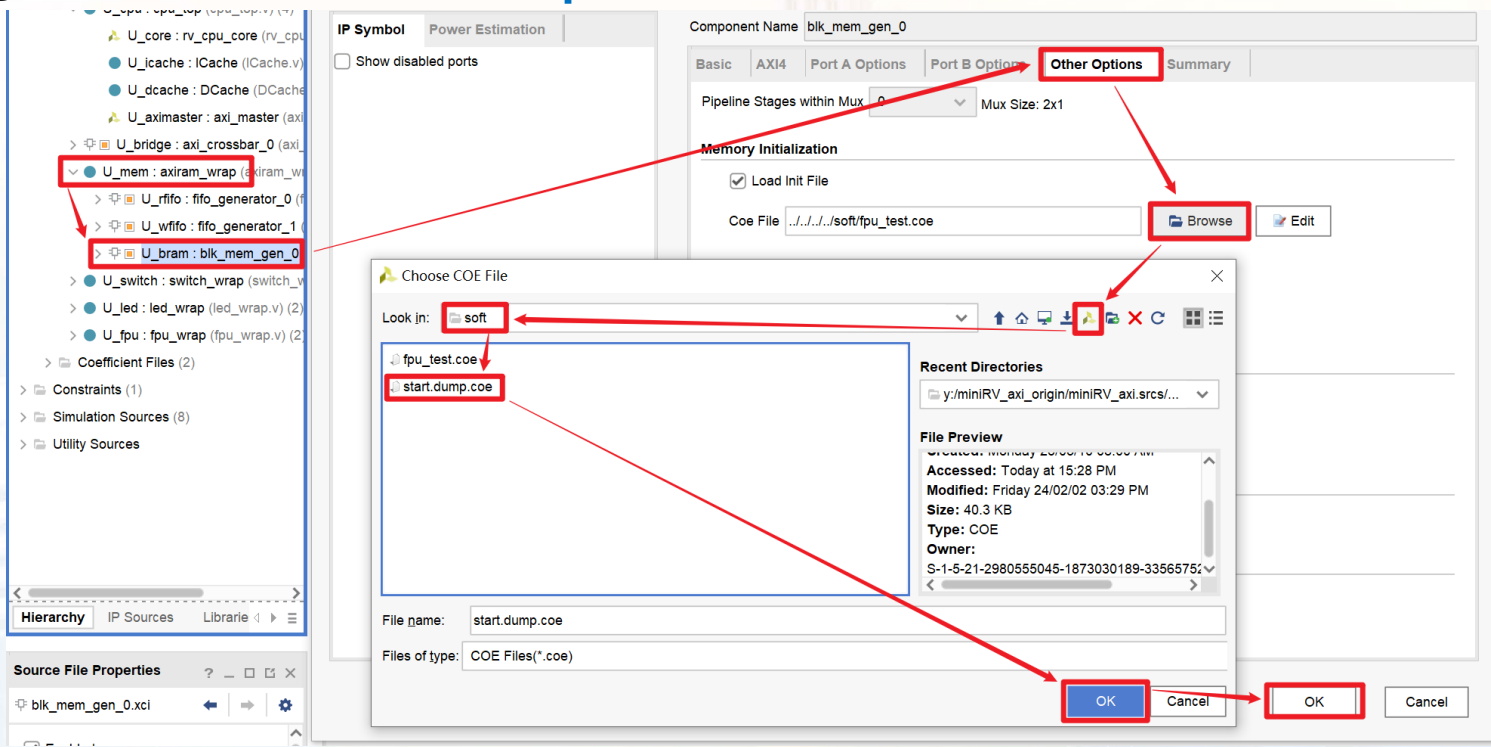
```
1|`timescale 1ns / 1ps  
2|`define TIME_OUT_THRESHOLD 160 // 改成320
```



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



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

- 若使用2018.3以上的版本，
需升级IP核

The screenshot displays the Vivado IDE interface with the following components:

- Hierarchy:** A tree view showing the project structure. The IP core `blk_mem_gen_1` is highlighted.
- Source File Properties:** A window showing details for `blk_mem_gen_1.xci`. It is marked as "IP is locked" and "Enabled".
- Source Node Properties:** A context menu is open for the selected IP, with the **Report IP Status** option highlighted.
- IP Status:** A table showing the status of various IP cores. The `blk_mem_gen_1` row is selected, and the **Upgrade Selected** button is highlighted.

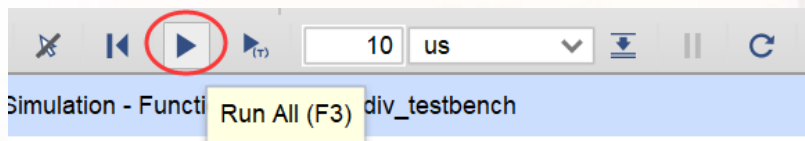
IP Status Table:

Source File	IP Status	Recommendation	Change Log	IP Name	Current Version	Recommended Version	License
axi_gpio_0	✓	IP revision change	Upgrade IP	AXI GPIO	2.0 (Rev. 20)	2.0 (Rev. 22)	Included
axi_gpio_1	✓	IP revision change	Upgrade IP	AXI GPIO	2.0 (Rev. 20)	2.0 (Rev. 22)	Included
axi_protocol_converter_0	✓	IP revision change	Upgrade IP	AXI Protocol Converter	2.1 (Rev. 18)	2.1 (Rev. 20)	Included
blk_mem_gen_0	✓	IP revision change	Upgrade IP	Block Memory Generator	8.4 (Rev. 2)	8.4 (Rev. 4)	Included
blk_mem_gen_1	✓	IP revision change	Upgrade IP	Block Memory Generator	8.4 (Rev. 2)	8.4 (Rev. 4)	Included
fifo_generator_0	✓	IP revision change	Upgrade IP	FIFO Generator	13.2 (Rev. 3)	13.2 (Rev. 5)	Included

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

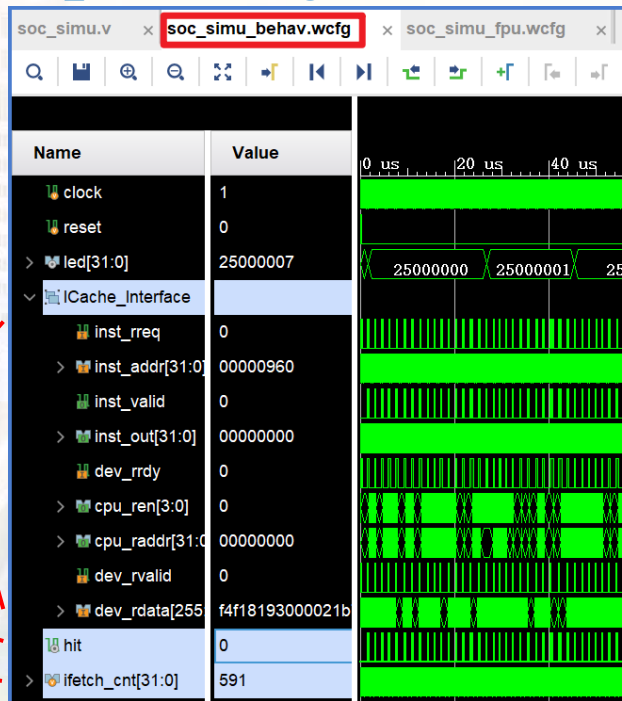
③ 点击“Run All”运行功能仿真，在soc_simu_behav.wcfg中查看波形



➤ 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

```
Tcl Console x Messages Log
[Icons]
----- test_01 -----
取指地址:      0x00000004
应取指令:      0x02500413
实际取到的指令: 0x0040006f
本次取指花费的时钟数: 1
取指出错, 请检查ICache逻辑
```

测试通过

```
Tcl Console x Messages Log
[Icons]
应取指令:      0x0080a023
实际取到的指令: 0x0080a023
本次取指花费的时钟数: 1
----- test_24 -----
[555.360us] test_24 passed!
All tests passed!
Total: 1968 instruction fetching, 26 memory access (18 RDs and 8 WRs)
ICache hit rate: 82.215%
```

测试程序总运行时间

总取指数

命中率

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

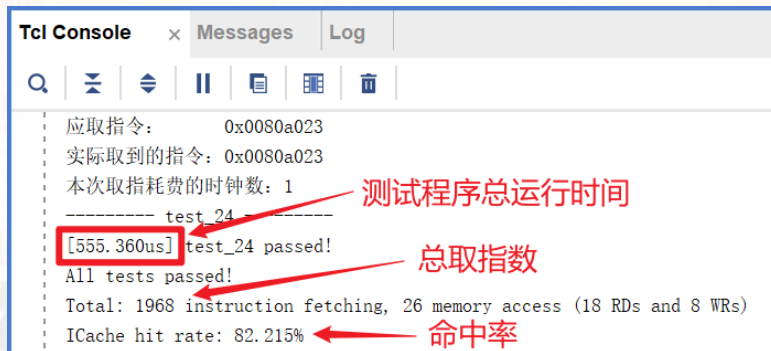
defines.vh

```
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：2.5分

- 提交内容

- ICache模块的.v源文件：1.5分
 - 实验报告（按模板完成）：4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