中国科学院计算技术研究所
INSTITUTE OF COMPUTING TECHNOLOGY ,CHINESE ACADEMY OF SCIENCES

中国开放指令生态（RISC-V）联盟
China RISC-V Alliance

鹏城实验室
Peng Cheng Laboratory

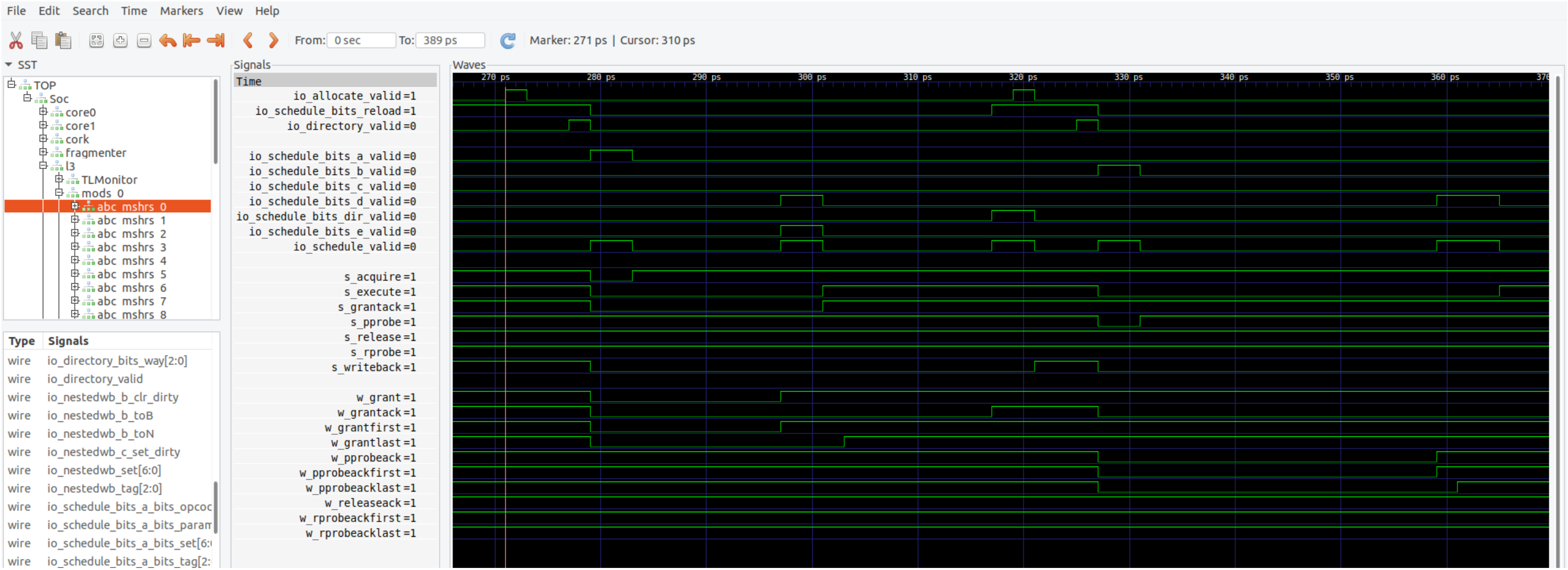
Waveform Terminator
填补底层波形与高层语义鸿沟的调试栈

蔺嘉炜 余子濠 王凯帆
中科院计算所

2021年6月23日

传统硬件仿真调试方法: 波形分析

- 分析波形的过程实际上是从波形中提取信息的过程
- 波形本身不包含语义信息，人工分析效率低、任务重



From: 0 sec To: 389 ps

Marker: 271 ps | Cursor: 310 ps

```
TOP
  Soc
    core0
    core1
    cork
    fragmenter
    l3
      TLMonitor
      mods_0
        abc_mshrs_0
        abc_mshrs_1
        abc_mshrs_2
        abc_mshrs_3
        abc_mshrs_4
        abc_mshrs_5
        abc_mshrs_6
        abc_mshrs_7
        abc_mshrs_8
```

Type	Signals
wire	io_directory_bits_way[2:0]
wire	io_directory_valid
wire	io_nestedwb_b_clr_dirty
wire	io_nestedwb_b_toB
wire	io_nestedwb_b_toN
wire	io_nestedwb_c_set_dirty
wire	io_nestedwb_set[6:0]
wire	io_nestedwb_tag[2:0]
wire	io_schedule_bits_a_bits_opcoc
wire	io_schedule_bits_a_bits_param
wire	io_schedule_bits_a_bits_set[6:
wire	io_schedule_bits_a_bits_tag[2:

```
Time
      io_allocate_valid=1
  io_schedule_bits_reload=1
     io_directory_valid=0

  io_schedule_bits_a_valid=0
  io_schedule_bits_b_valid=0
  io_schedule_bits_c_valid=0
  io_schedule_bits_d_valid=0
io_schedule_bits_dir_valid=0
  io_schedule_bits_e_valid=0
      io_schedule_valid=0

              s_acquire=1
              s_execute=1
             s_grantack=1
               s_pprobe=1
              s_release=1
               s_rprobe=1
             s_writeback=1

                w_grant=1
             w_grantack=1
           w_grantfirst=1
            w_grantlast=1
            w_pprobeack=1
       w_pprobeackfirst=1
        w_pprobeacklast=1
           w_releaseack=1
       w_rprobeackfirst=1
        w_rprobeacklast=1
```

Waveform Terminator: 新型硬件调试栈

- 我们设计了一套工具，能够将高层语义信息从波形中提取出来

日志分析　对事件日志进行高层次的语义分析和检查

Firrtl Transform　用于将Chisel中开发者关心的事件自动转换成 ”Xiang” 语言描述的转换规则

Xiang语言　自定义的DSL “Xiang” 语言，用于描述波形信息到事件日志的转换规则

波形文件Parser　解析波形文件

Waveform Terminator: 香语言

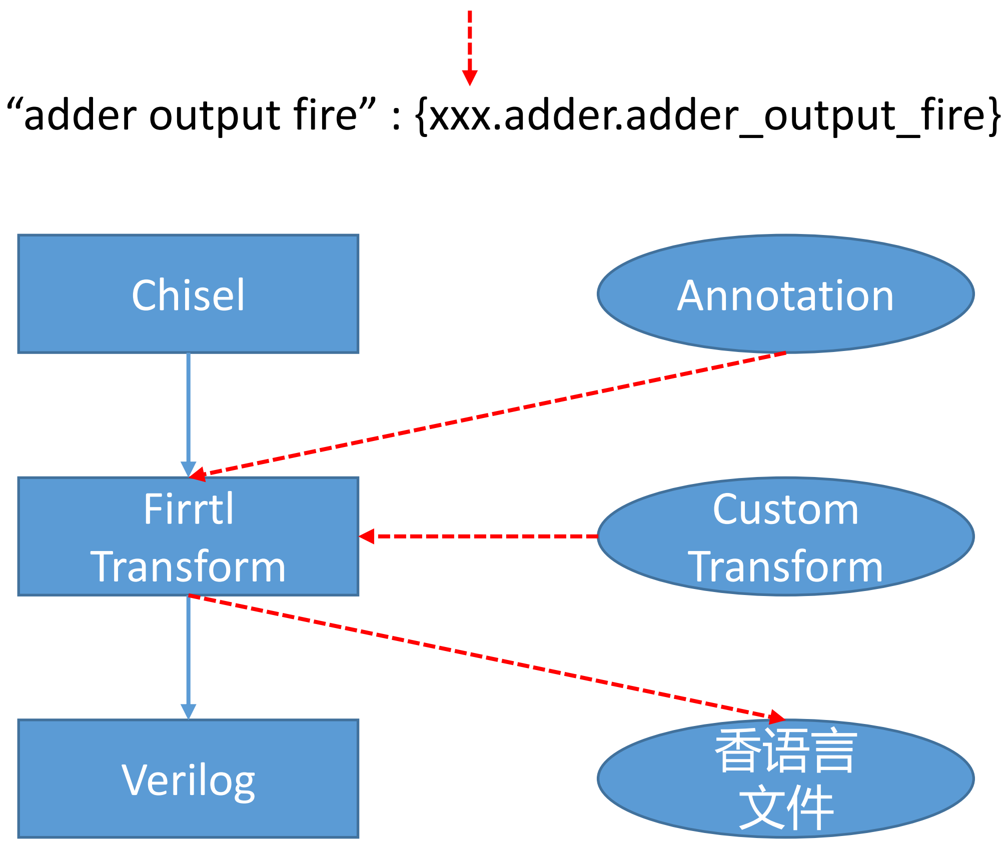
- 一种DSL，用于描述底层波形到高层语义的转换规则
- 将基于波形的调试转换成基于事件的调试
- Waveform Terminator解析香语言并根据对应规则从波形中提取信息
- 语法简单易用
  - “name”: {expression}
  - 示例: “adder output fire” : { adder.io.out.valid && adder.io.out.ready }

Waveform Terminator: 香语言

- 大型设计中，模块层次结构复杂
  - Chisel中的匿名信号生成Verilog后人工追踪困难
  - 手写香语言工作量太大
- 因此我们设计了两种辅助工具
  - Custom Firrtl Transform
  - Xiang Editor - 可视化编写界面
    - 提供信号、模块路径补全功能

Log.add(“adder output fire”, io.out.fire())

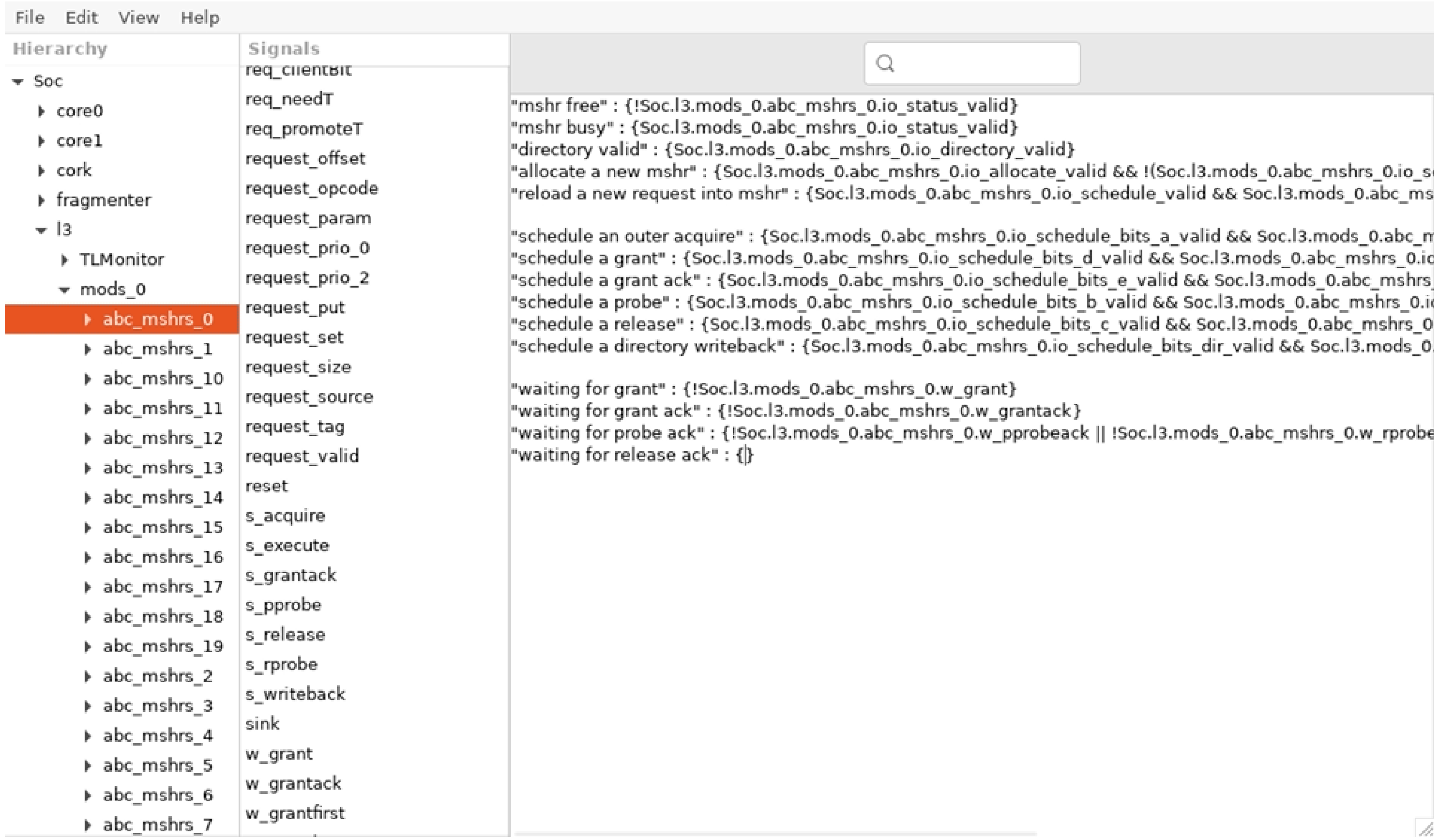
“adder output fire” : {xxx.adder.adder_output_fire}



Custom Firrtl Transform 工作流

Xiang Editor

- 解析波形文件
- 提供设计的层次结构及每个子模块中的信号
- 提供自动信号补全功能
- 使修改、编辑香语言更加方便

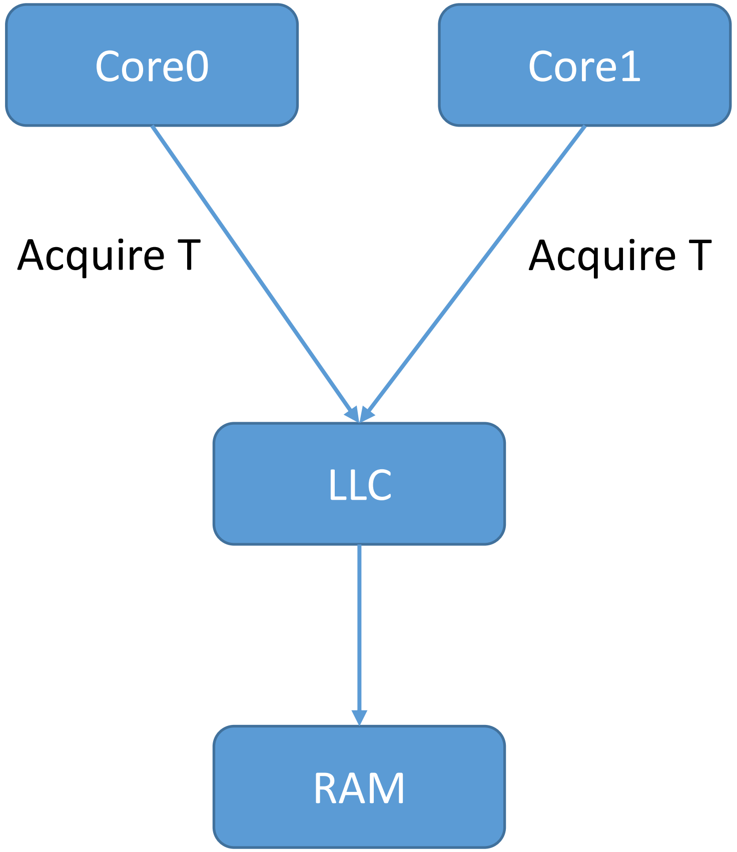


```
"mshr free" : {!Soc.l3.mods_0.abc_mshrs_0.io_status_valid}
"mshr busy" : {Soc.l3.mods_0.abc_mshrs_0.io_status_valid}
"directory valid" : {Soc.l3.mods_0.abc_mshrs_0.io_directory_valid}
"allocate a new mshr" : {Soc.l3.mods_0.abc_mshrs_0.io_allocate_valid && !(Soc.l3.mods_0.abc_mshrs_0.io_s
"reload a new request into mshr" : {Soc.l3.mods_0.abc_mshrs_0.io_schedule_valid && Soc.l3.mods_0.abc_ms

"schedule an outer acquire" : {Soc.l3.mods_0.abc_mshrs_0.io_schedule_bits_a_valid && Soc.l3.mods_0.abc_m
"schedule a grant" : {Soc.l3.mods_0.abc_mshrs_0.io_schedule_bits_d_valid && Soc.l3.mods_0.abc_mshrs_0.io
"schedule a grant ack" : {Soc.l3.mods_0.abc_mshrs_0.io_schedule_bits_e_valid && Soc.l3.mods_0.abc_mshrs_
"schedule a probe" : {Soc.l3.mods_0.abc_mshrs_0.io_schedule_bits_b_valid && Soc.l3.mods_0.abc_mshrs_0.io
"schedule a release" : {Soc.l3.mods_0.abc_mshrs_0.io_schedule_bits_c_valid && Soc.l3.mods_0.abc_mshrs_0
"schedule a directory writeback" : {Soc.l3.mods_0.abc_mshrs_0.io_schedule_bits_dir_valid && Soc.l3.mods_0.

"waiting for grant" : {!Soc.l3.mods_0.abc_mshrs_0.w_grant}
"waiting for grant ack" : {!Soc.l3.mods_0.abc_mshrs_0.w_grantack}
"waiting for probe ack" : {!Soc.l3.mods_0.abc_mshrs_0.w_pprobeack || !Soc.l3.mods_0.abc_mshrs_0.w_rprobe
"waiting for release ack" : {}
```

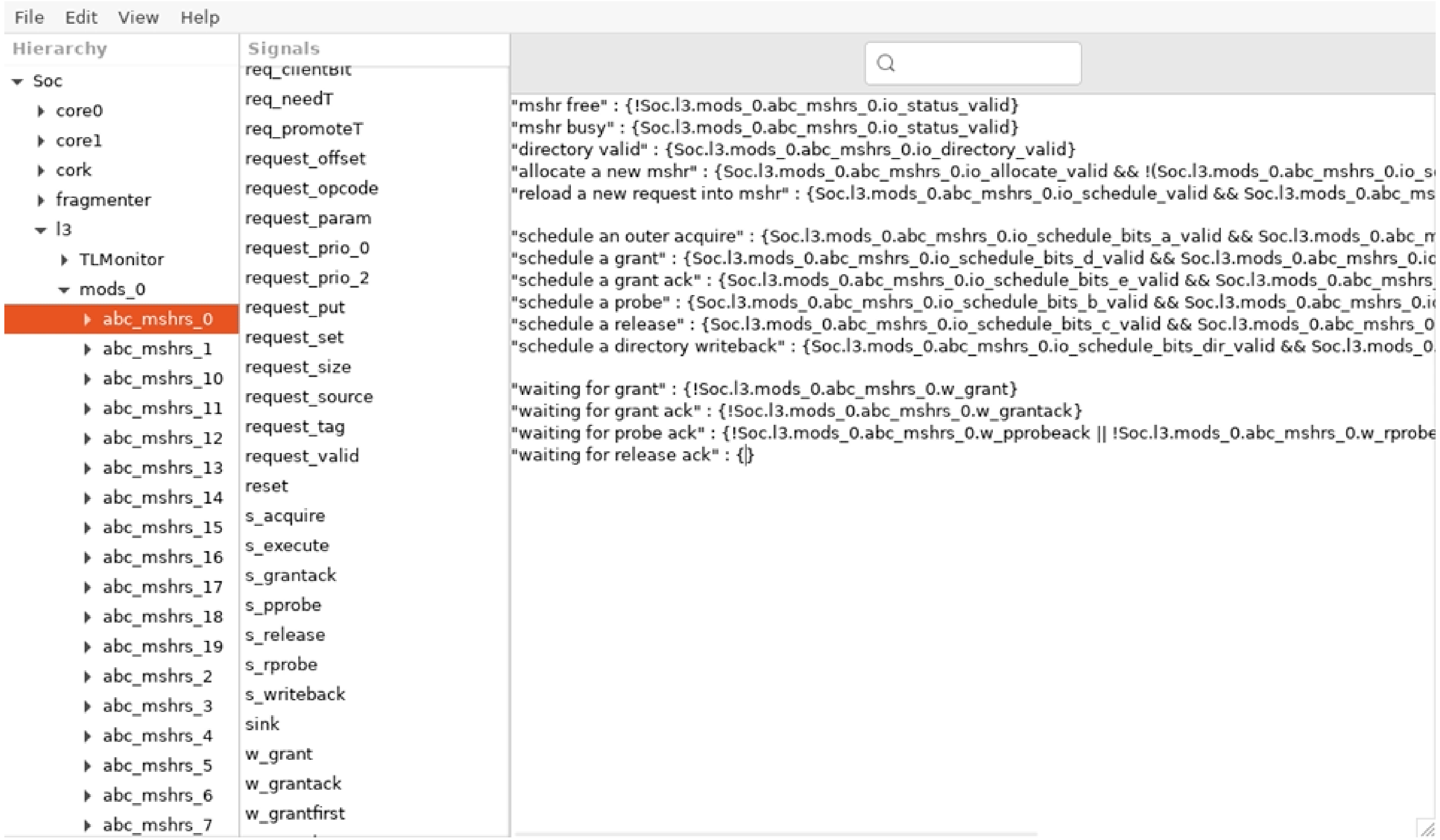
使用Xiang Editor为Cache的工作过程编写事件转换规则

Waveform Terminator: 应用示例

- 以分析基于Tilelink协议的LLC中一个MSHR对两个请求的处理流程为例
- 系统连接简化模型如右图所示
- Core0、Core1向LLC Acquire同一个Block，均要求T权限(写权限)
- 初始状态时Core0、Core1、LLC中均无该Block



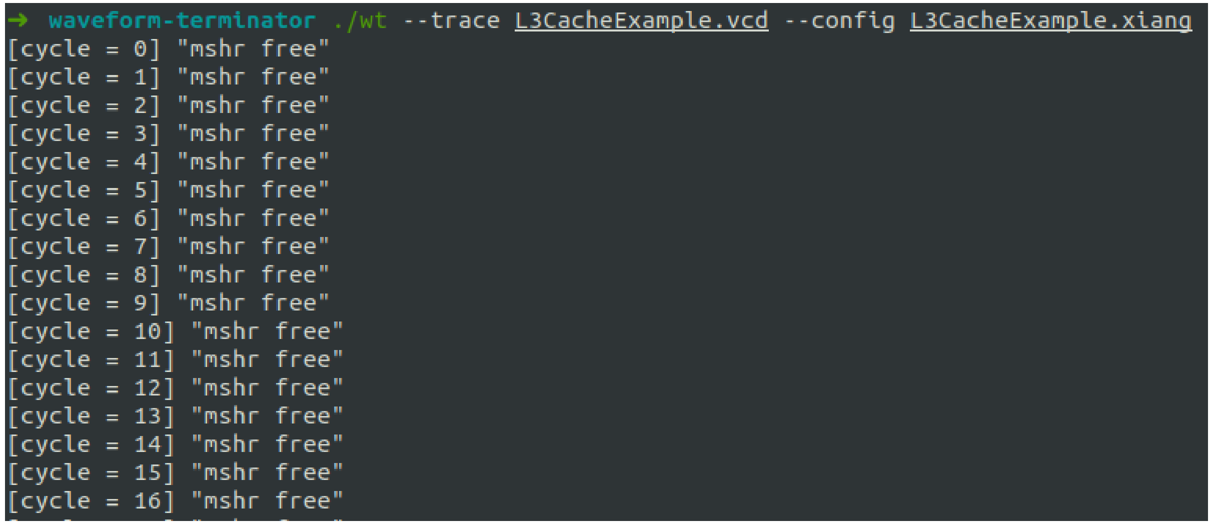
Waveform Terminator: 应用示例—编写转换规则



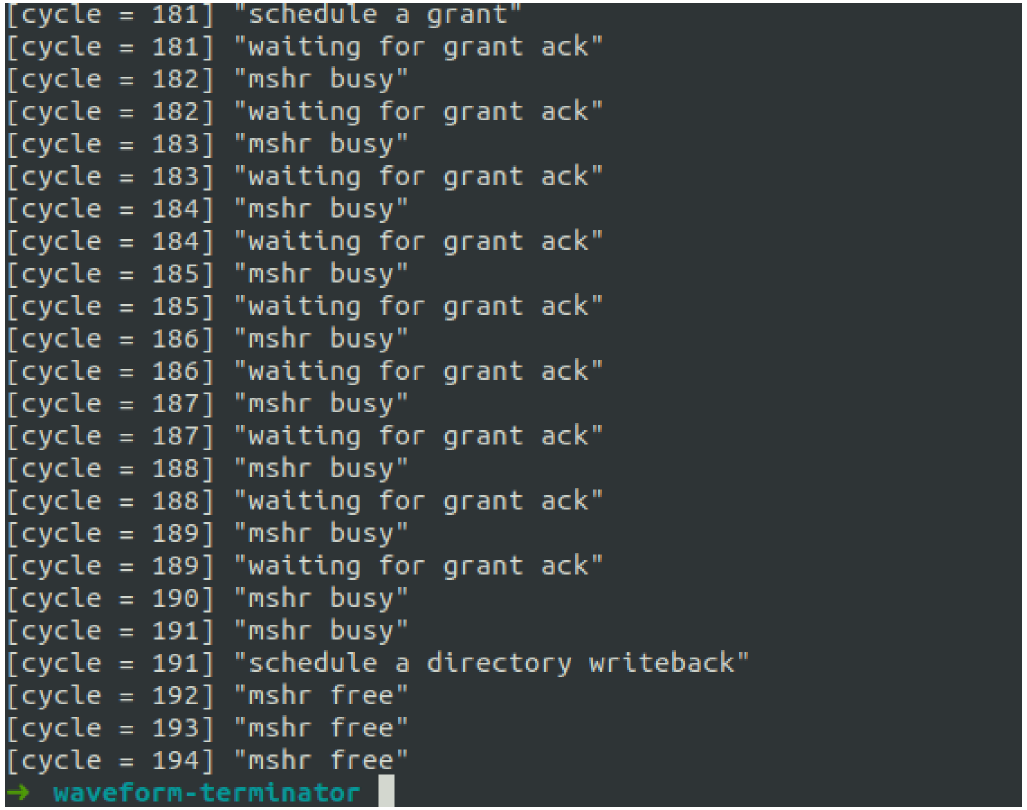
```
"mshr free" : {!Soc.l3.mods_0.abc_mshrs_0.io_status_valid}
"mshr busy" : {Soc.l3.mods_0.abc_mshrs_0.io_status_valid}
"directory valid" : {Soc.l3.mods_0.abc_mshrs_0.io_directory_valid}
"allocate a new mshr" : {Soc.l3.mods_0.abc_mshrs_0.io_allocate_valid && !(Soc.l3.mods_0.abc_mshrs_0.io_s
"reload a new request into mshr" : {Soc.l3.mods_0.abc_mshrs_0.io_schedule_valid && Soc.l3.mods_0.abc_ms

"schedule an outer acquire" : {Soc.l3.mods_0.abc_mshrs_0.io_schedule_bits_a_valid && Soc.l3.mods_0.abc_m
"schedule a grant" : {Soc.l3.mods_0.abc_mshrs_0.io_schedule_bits_d_valid && Soc.l3.mods_0.abc_mshrs_0.i
"schedule a grant ack" : {Soc.l3.mods_0.abc_mshrs_0.io_schedule_bits_e_valid && Soc.l3.mods_0.abc_mshrs_
"schedule a probe" : {Soc.l3.mods_0.abc_mshrs_0.io_schedule_bits_b_valid && Soc.l3.mods_0.abc_mshrs_0.i
"schedule a release" : {Soc.l3.mods_0.abc_mshrs_0.io_schedule_bits_c_valid && Soc.l3.mods_0.abc_mshrs_0
"schedule a directory writeback" : {Soc.l3.mods_0.abc_mshrs_0.io_schedule_bits_dir_valid && Soc.l3.mods_0.

"waiting for grant" : {!Soc.l3.mods_0.abc_mshrs_0.w_grant}
"waiting for grant ack" : {!Soc.l3.mods_0.abc_mshrs_0.w_grantack}
"waiting for probe ack" : {!Soc.l3.mods_0.abc_mshrs_0.w_pprobeack || !Soc.l3.mods_0.abc_mshrs_0.w_rprobe
"waiting for release ack" : {}
```

Waveform Terminator: 应用示例—提取Log

- 输入：一个波形文件，一个香语言转换规则描述文件
- 输出：从波形中提取的Log

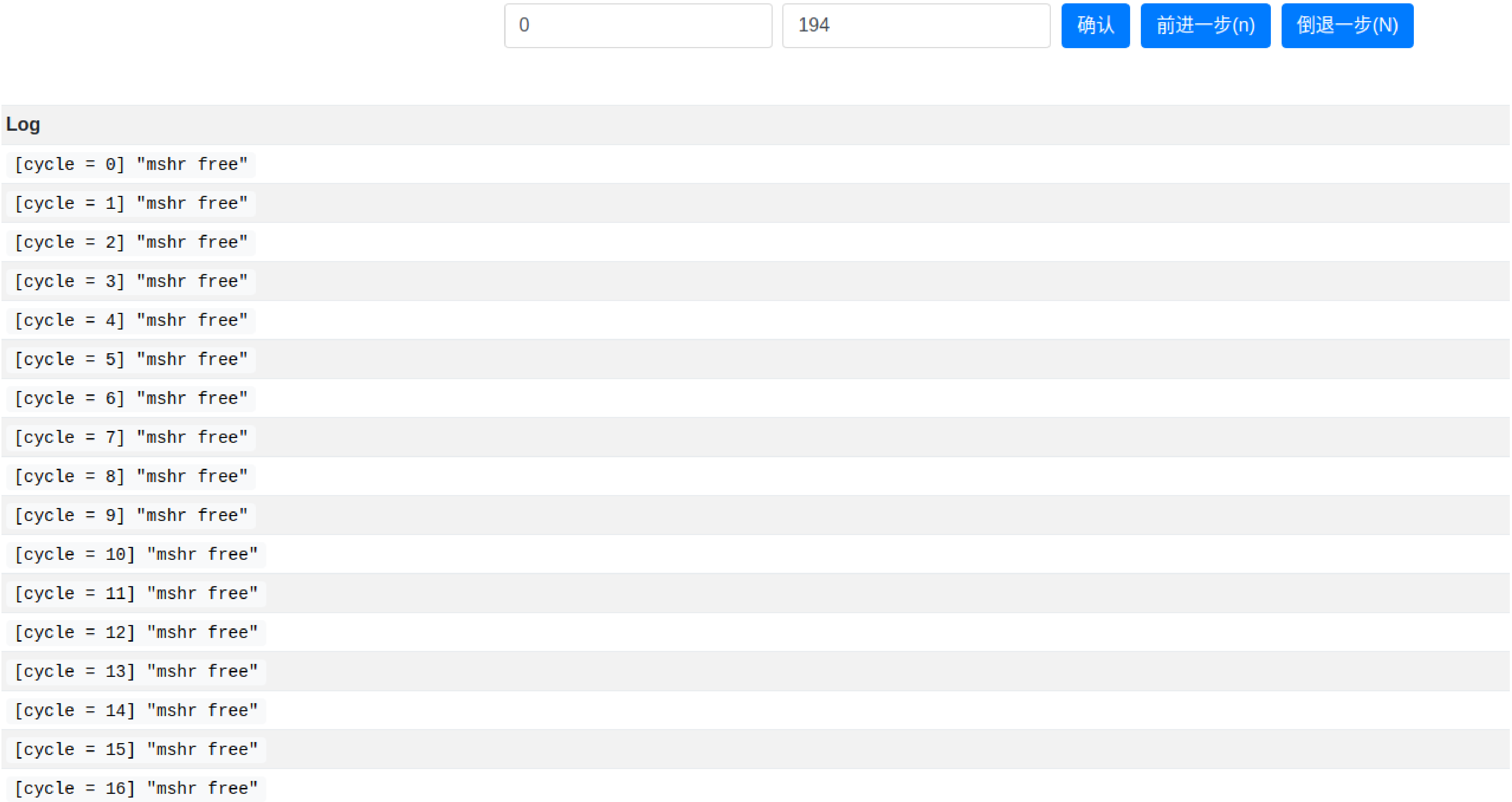


```
➜ waveform-terminator ./wt --trace L3CacheExample.vcd --config L3CacheExample.xiang
[cycle = 0] "mshr free"
[cycle = 1] "mshr free"
[cycle = 2] "mshr free"
[cycle = 3] "mshr free"
[cycle = 4] "mshr free"
[cycle = 5] "mshr free"
[cycle = 6] "mshr free"
[cycle = 7] "mshr free"
[cycle = 8] "mshr free"
[cycle = 9] "mshr free"
[cycle = 10] "mshr free"
[cycle = 11] "mshr free"
[cycle = 12] "mshr free"
[cycle = 13] "mshr free"
[cycle = 14] "mshr free"
[cycle = 15] "mshr free"
[cycle = 16] "mshr free"
```



```
[cycle = 181] "schedule a grant"
[cycle = 181] "waiting for grant ack"
[cycle = 182] "mshr busy"
[cycle = 182] "waiting for grant ack"
[cycle = 183] "mshr busy"
[cycle = 183] "waiting for grant ack"
[cycle = 184] "mshr busy"
[cycle = 184] "waiting for grant ack"
[cycle = 185] "mshr busy"
[cycle = 185] "waiting for grant ack"
[cycle = 186] "mshr busy"
[cycle = 186] "waiting for grant ack"
[cycle = 187] "mshr busy"
[cycle = 187] "waiting for grant ack"
[cycle = 188] "mshr busy"
[cycle = 188] "waiting for grant ack"
[cycle = 189] "mshr busy"
[cycle = 189] "waiting for grant ack"
[cycle = 190] "mshr busy"
[cycle = 191] "mshr busy"
[cycle = 191] "schedule a directory writeback"
[cycle = 192] "mshr free"
[cycle = 193] "mshr free"
[cycle = 194] "mshr free"
➜ waveform-terminator
```

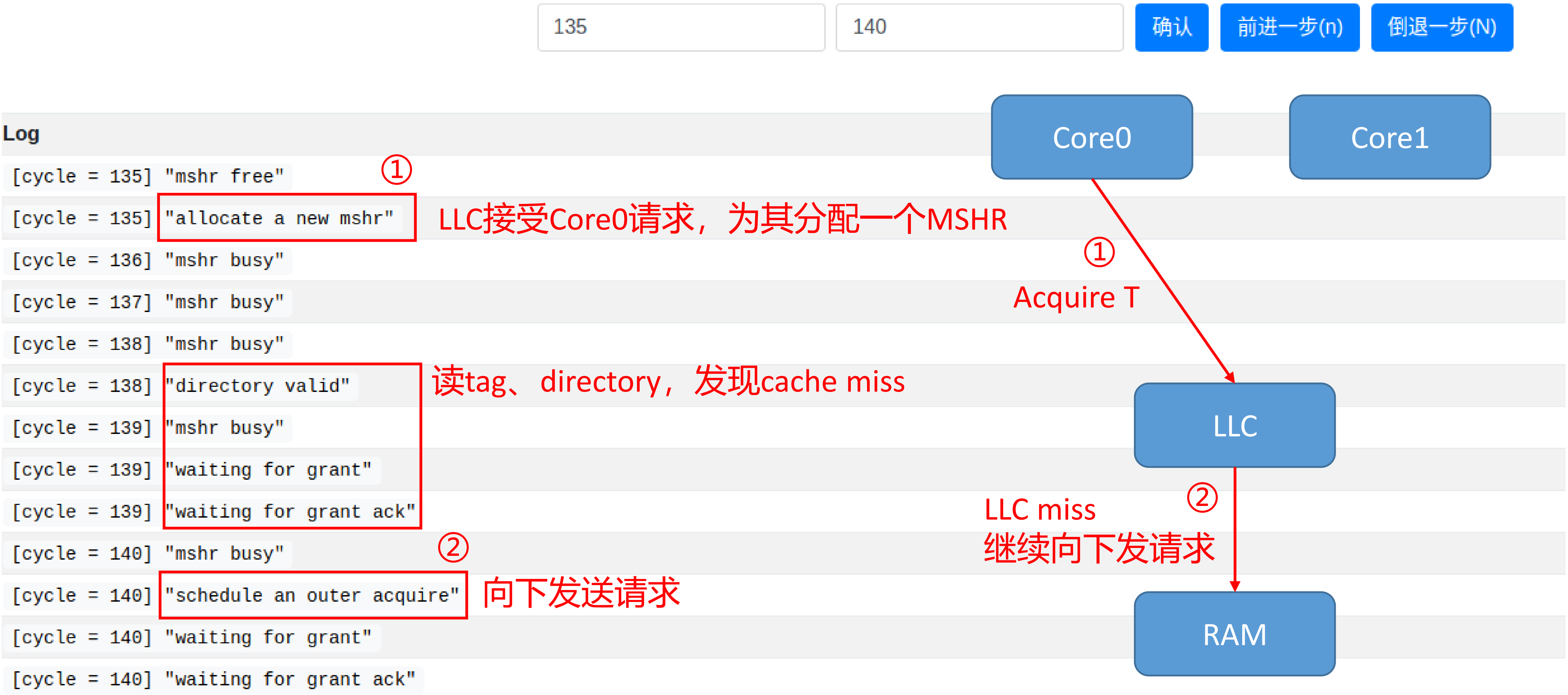
Waveform Terminator: 应用示例—Log分析



Log

```
[cycle = 0] "mshr free"
[cycle = 1] "mshr free"
[cycle = 2] "mshr free"
[cycle = 3] "mshr free"
[cycle = 4] "mshr free"
[cycle = 5] "mshr free"
[cycle = 6] "mshr free"
[cycle = 7] "mshr free"
[cycle = 8] "mshr free"
[cycle = 9] "mshr free"
[cycle = 10] "mshr free"
[cycle = 11] "mshr free"
[cycle = 12] "mshr free"
[cycle = 13] "mshr free"
[cycle = 14] "mshr free"
[cycle = 15] "mshr free"
[cycle = 16] "mshr free"
```

Waveform Terminator: 应用示例—Log分析



Log

```
[cycle = 135] "mshr free"
[cycle = 135] "allocate a new mshr"
[cycle = 136] "mshr busy"
[cycle = 137] "mshr busy"
[cycle = 138] "mshr busy"
[cycle = 138] "directory valid"
[cycle = 139] "mshr busy"
[cycle = 139] "waiting for grant"
[cycle = 139] "waiting for grant ack"
[cycle = 140] "mshr busy"
[cycle = 140] "schedule an outer acquire"
[cycle = 140] "waiting for grant"
[cycle = 140] "waiting for grant ack"
```

① LLC接受Core0请求，为其分配一个MSHR

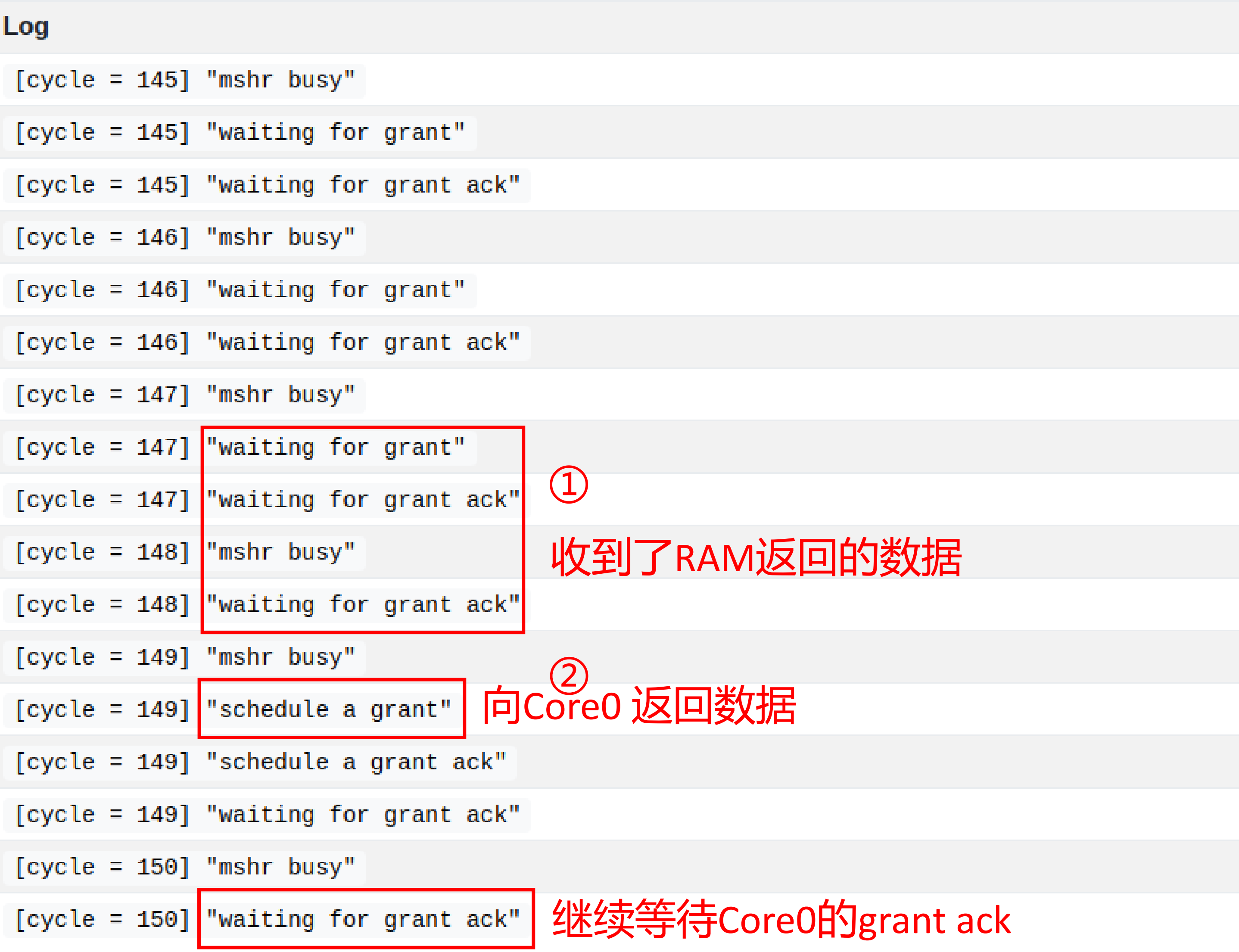
读tag、directory，发现cache miss

② 向下发送请求

① Acquire T

② LLC miss
继续向下发请求

Waveform Terminator: 应用示例—Log分析

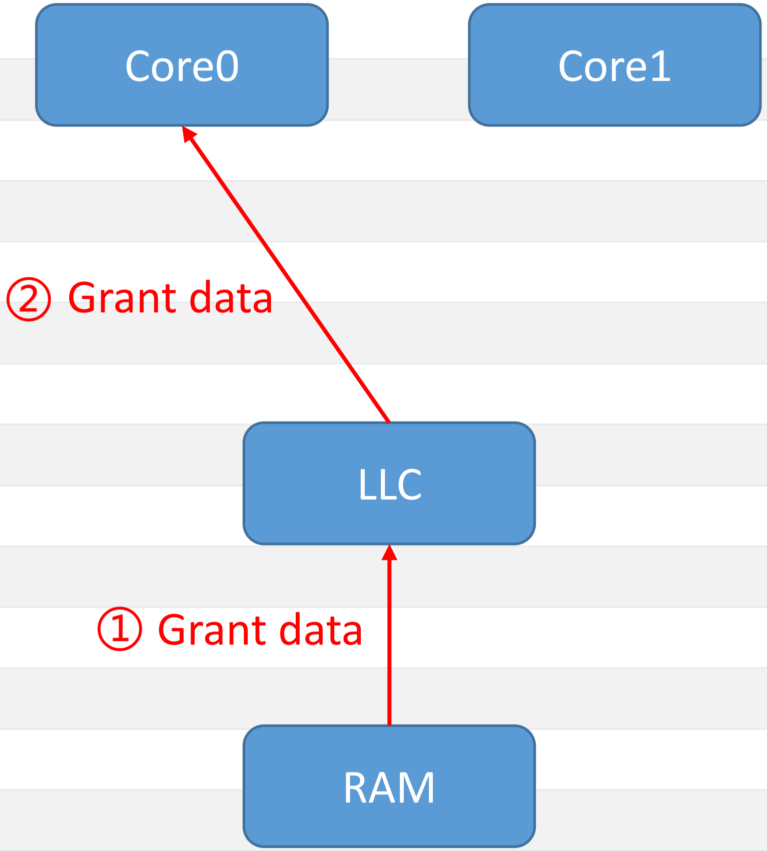


```
Log
[cycle = 145] "mshr busy"
[cycle = 145] "waiting for grant"
[cycle = 145] "waiting for grant ack"
[cycle = 146] "mshr busy"
[cycle = 146] "waiting for grant"
[cycle = 146] "waiting for grant ack"
[cycle = 147] "mshr busy"
[cycle = 147] "waiting for grant"
[cycle = 147] "waiting for grant ack"
[cycle = 148] "mshr busy"
[cycle = 148] "waiting for grant ack"
[cycle = 149] "mshr busy"
[cycle = 149] "schedule a grant"
[cycle = 149] "schedule a grant ack"
[cycle = 149] "waiting for grant ack"
[cycle = 150] "mshr busy"
[cycle = 150] "waiting for grant ack"
```

①
收到了RAM返回的数据

②
向Core0 返回数据

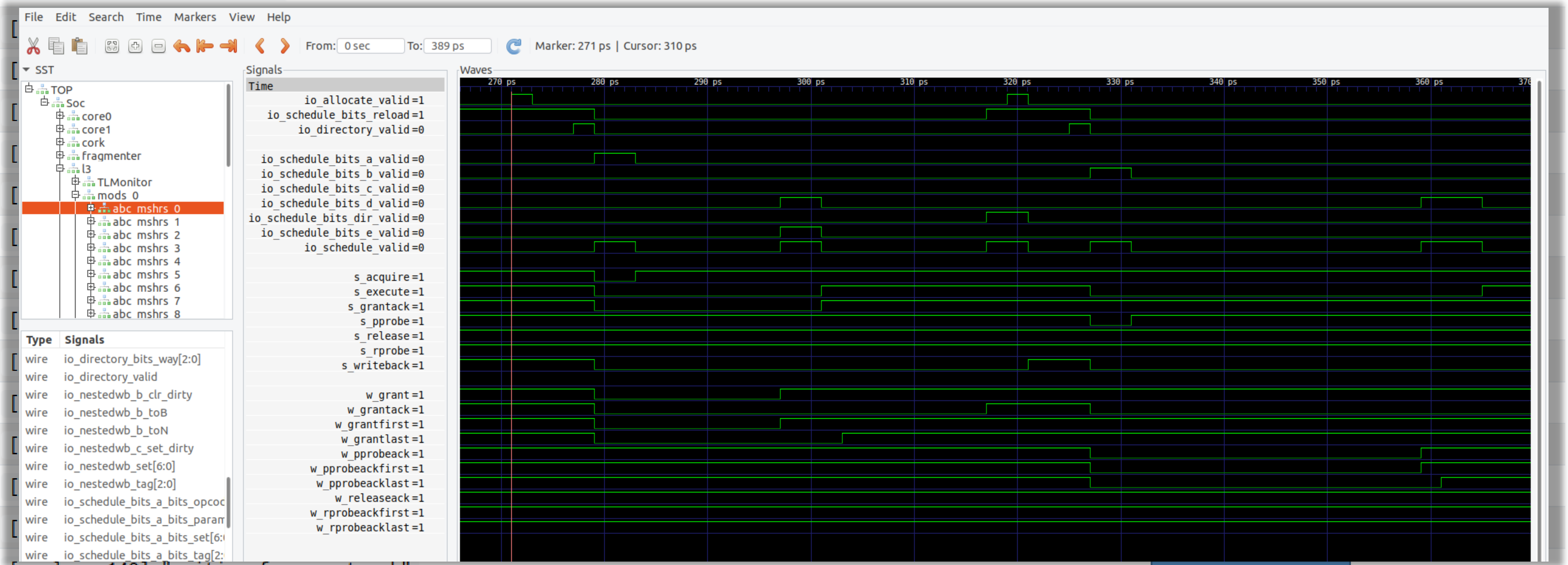
继续等待Core0的grant ack



Waveform Terminator: 与波形调试的对比

Log

```
[cycle = 145] "mshr busy"
```



Type	Signals
wire	io_directory_bits_way[2:0]
wire	io_directory_valid
wire	io_nestedwb_b_clr_dirty
wire	io_nestedwb_b_toB
wire	io_nestedwb_b_toN
wire	io_nestedwb_c_set_dirty
wire	io_nestedwb_set[6:0]
wire	io_nestedwb_tag[2:0]
wire	io_schedule_bits_a_bits_opcoc
wire	io_schedule_bits_a_bits_param
wire	io_schedule_bits_a_bits_set[6:
wire	io_schedule_bits_a_bits_tag[2:

```
Time
io_allocate_valid=1
io_schedule_bits_reload=1
io_directory_valid=0

io_schedule_bits_a_valid=0
io_schedule_bits_b_valid=0
io_schedule_bits_c_valid=0
io_schedule_bits_d_valid=0
io_schedule_bits_dir_valid=0
io_schedule_bits_e_valid=0
io_schedule_valid=0

s_acquire=1
s_execute=1
s_grantack=1
s_pprobe=1
s_release=1
s_rprobe=1
s_writeback=1

w_grant=1
w_grantack=1
w_grantfirst=1
w_grantlast=1
w_pprobeack=1
w_pprobeackfirst=1
w_pprobeacklast=1
w_releaseack=1
w_rprobeackfirst=1
w_rprobeacklast=1
```

```
[cycle = 149] "waiting for grant ack"
[cycle = 150] "mshr busy"
[cycle = 150] "waiting for grant ack"
```

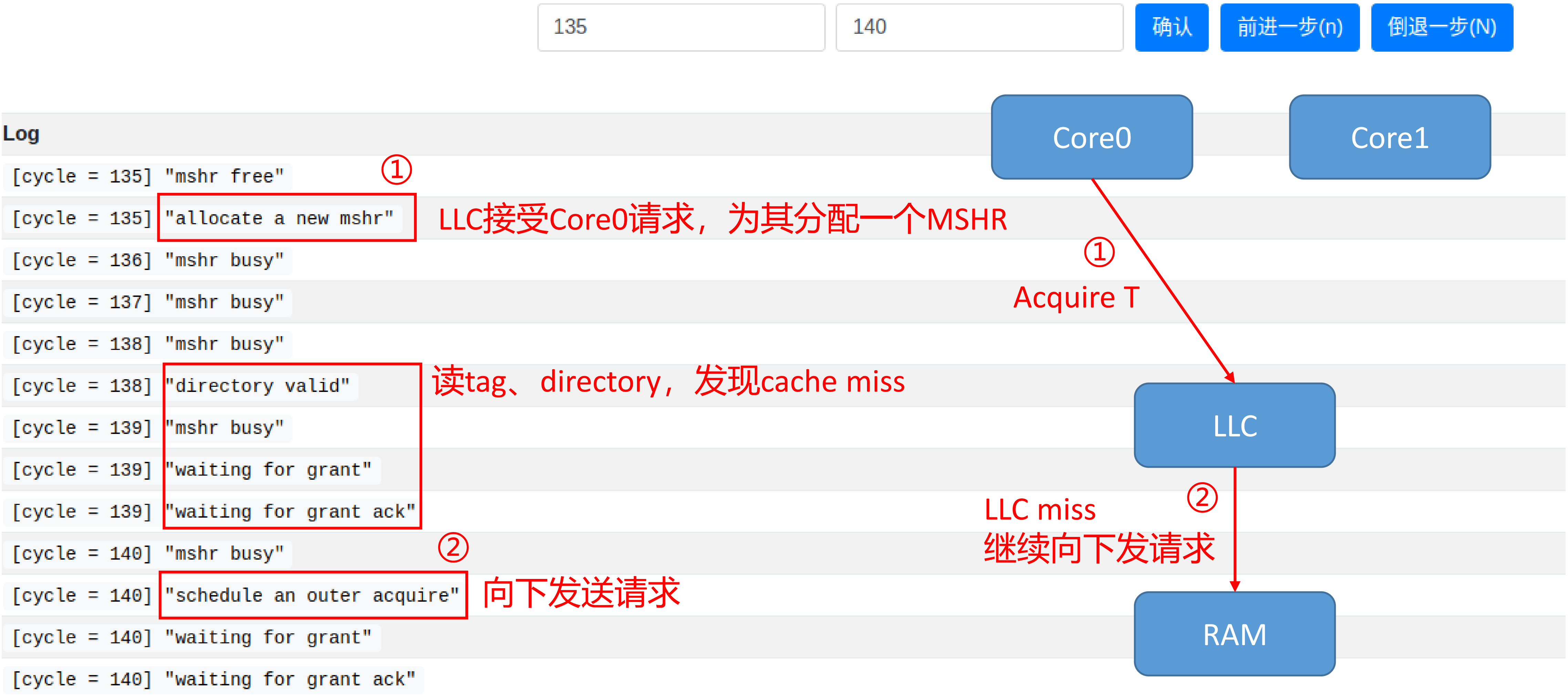
RAM

Waveform Terminator: 问题讨论

- 为何不使用fwrite/display/printf?
  - 每次修改事务转换规则后需要重新仿真
  - 对于使用Chisel开发的电路，修改/添加printf后还需要重新生成Verilog

敬请批评指正！

Waveform Terminator: 应用示例—Log分析



Log

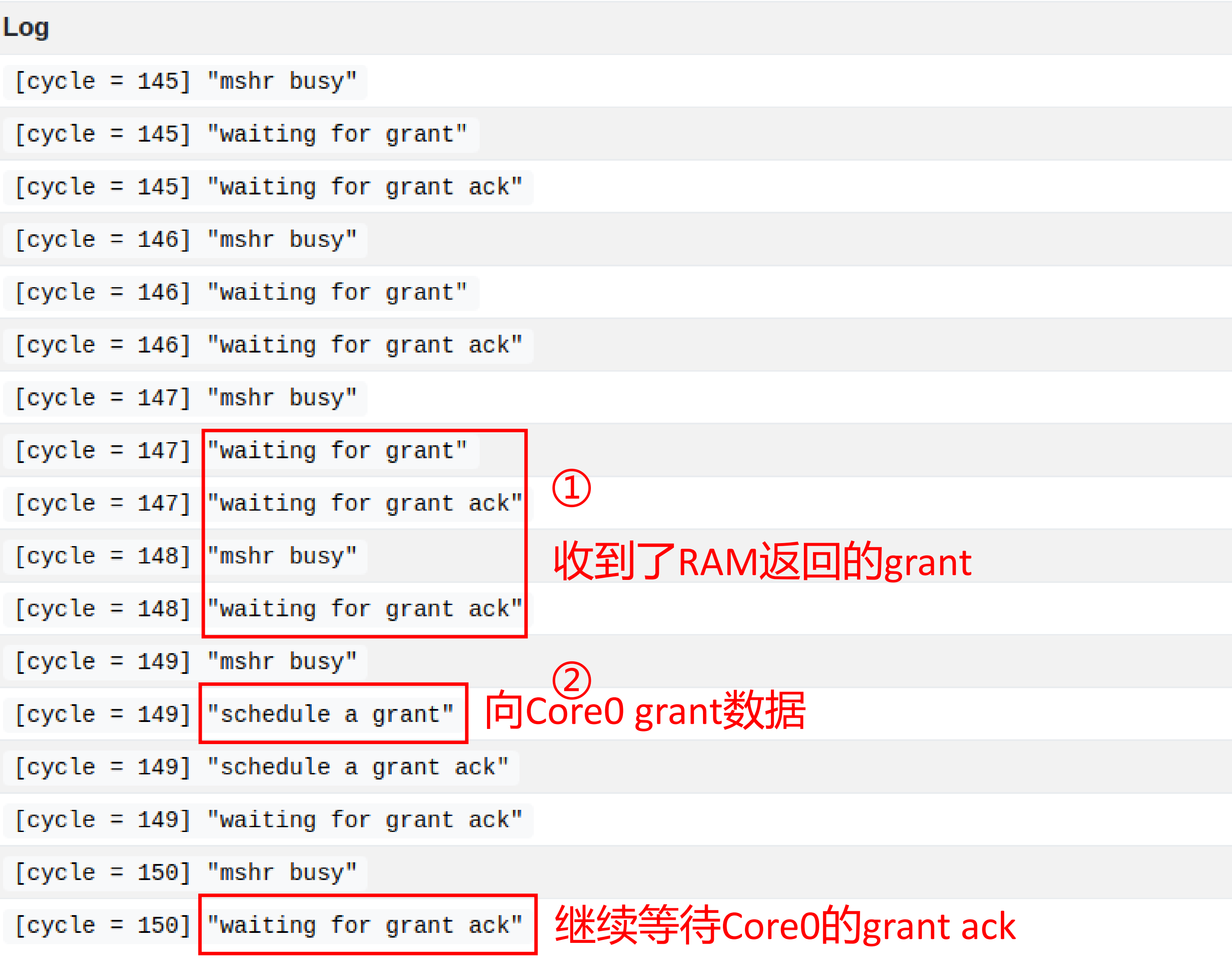
```
[cycle = 135] "mshr free"
[cycle = 135] "allocate a new mshr"
[cycle = 136] "mshr busy"
[cycle = 137] "mshr busy"
[cycle = 138] "mshr busy"
[cycle = 138] "directory valid"
[cycle = 139] "mshr busy"
[cycle = 139] "waiting for grant"
[cycle = 139] "waiting for grant ack"
[cycle = 140] "mshr busy"
[cycle = 140] "schedule an outer acquire"
[cycle = 140] "waiting for grant"
[cycle = 140] "waiting for grant ack"
```

① LLC接受Core0请求，为其分配一个MSHR

读tag、directory，发现cache miss

② 向下发送请求

Waveform Terminator: 应用示例—Log分析



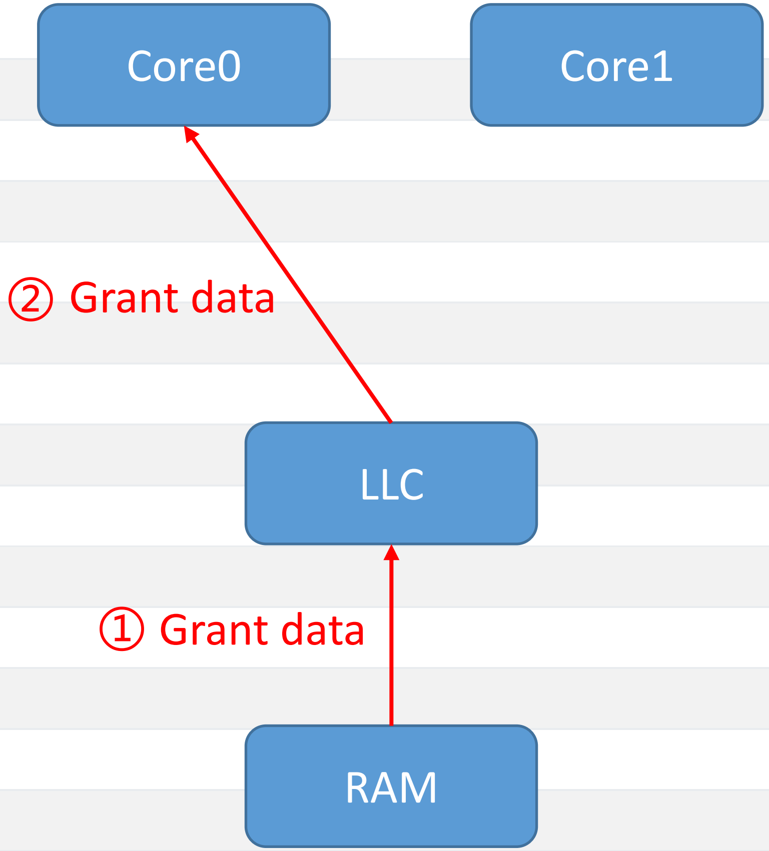
Log

```
[cycle = 145] "mshr busy"
[cycle = 145] "waiting for grant"
[cycle = 145] "waiting for grant ack"
[cycle = 146] "mshr busy"
[cycle = 146] "waiting for grant"
[cycle = 146] "waiting for grant ack"
[cycle = 147] "mshr busy"
[cycle = 147] "waiting for grant"
[cycle = 147] "waiting for grant ack"
[cycle = 148] "mshr busy"
[cycle = 148] "waiting for grant ack"
[cycle = 149] "mshr busy"
[cycle = 149] "schedule a grant"
[cycle = 149] "schedule a grant ack"
[cycle = 149] "waiting for grant ack"
[cycle = 150] "mshr busy"
[cycle = 150] "waiting for grant ack"
```

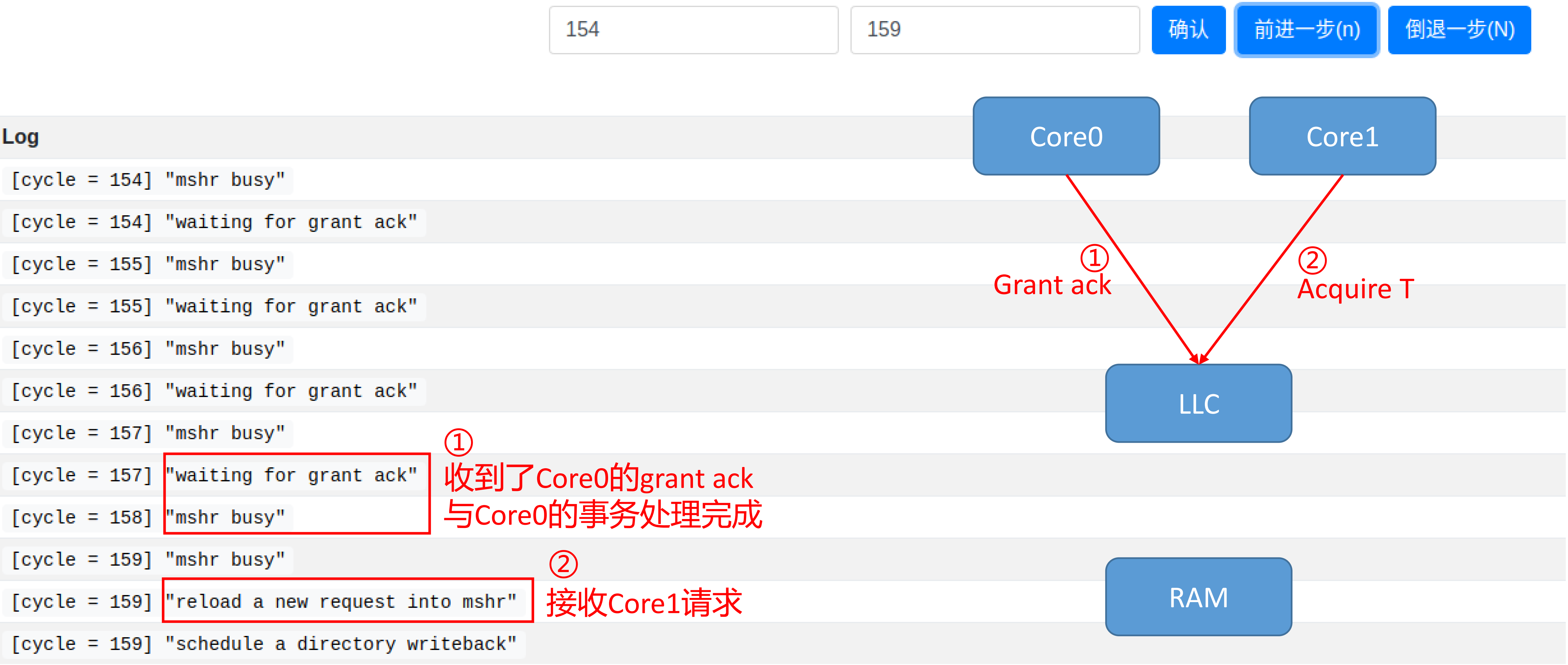
① 收到了RAM返回的grant

② 向Core0 grant数据

继续等待Core0的grant ack



Waveform Terminator: 应用示例—Log分析



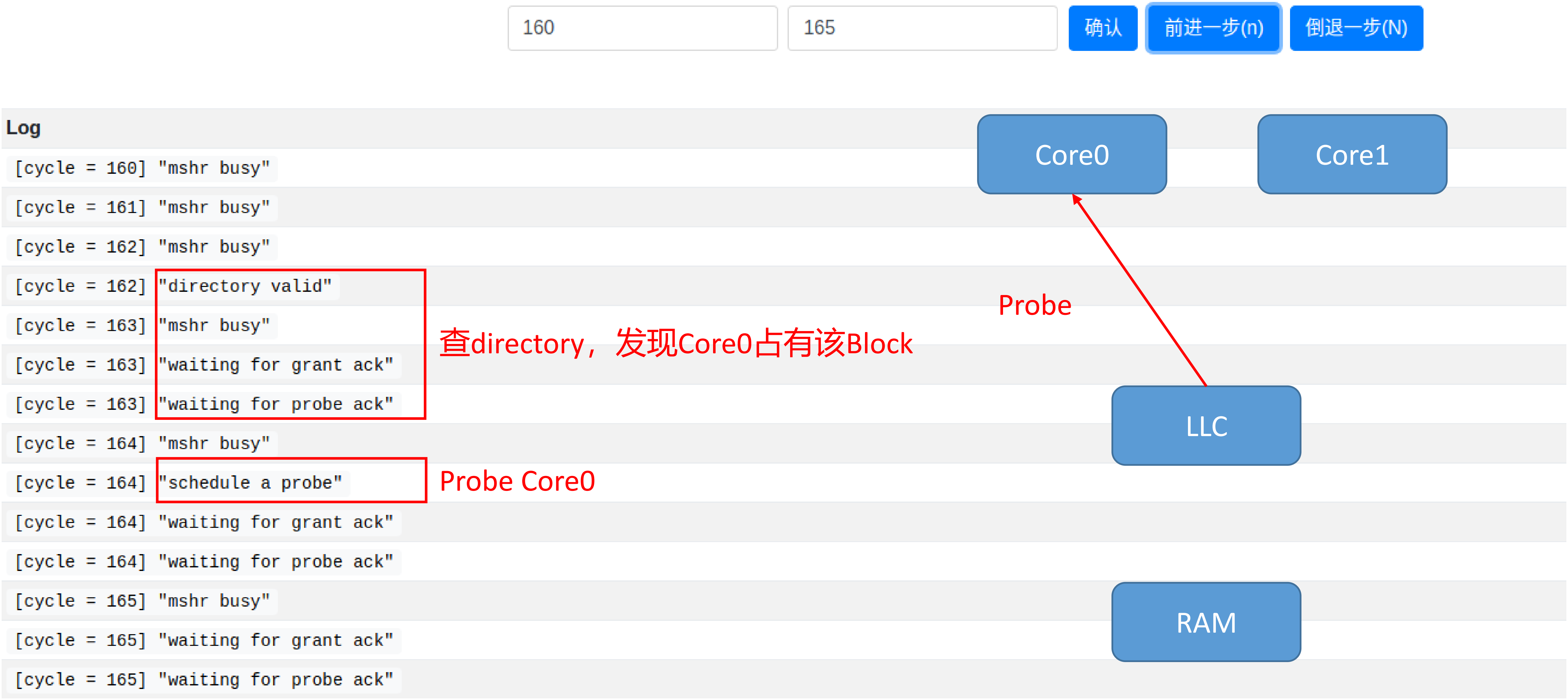
Log

```
[cycle = 154] "mshr busy"
[cycle = 154] "waiting for grant ack"
[cycle = 155] "mshr busy"
[cycle = 155] "waiting for grant ack"
[cycle = 156] "mshr busy"
[cycle = 156] "waiting for grant ack"
[cycle = 157] "mshr busy"
[cycle = 157] "waiting for grant ack"
[cycle = 158] "mshr busy"
[cycle = 159] "mshr busy"
[cycle = 159] "reload a new request into mshr"
[cycle = 159] "schedule a directory writeback"
```

① 收到了Core0的grant ack
与Core0的事务处理完成

② 接收Core1请求

Waveform Terminator: 应用示例—Log分析



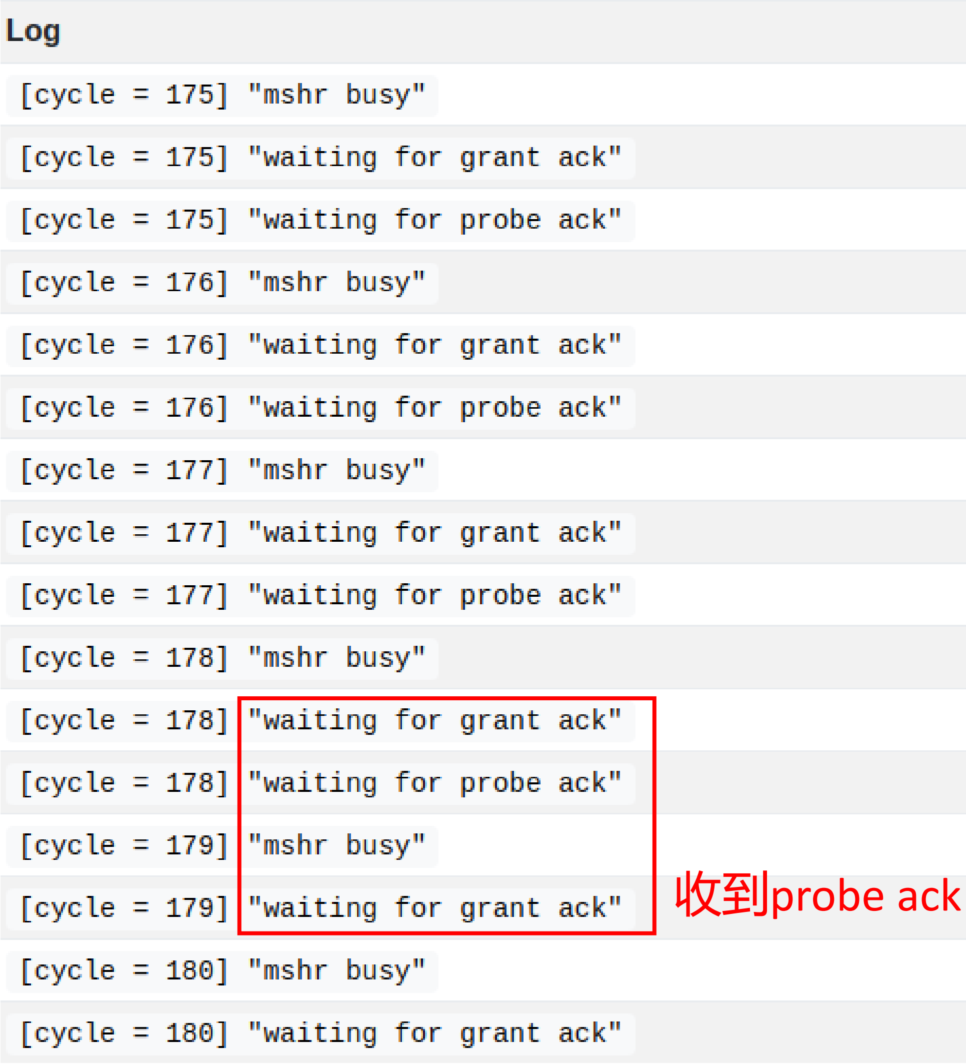
Log

```
[cycle = 160] "mshr busy"
[cycle = 161] "mshr busy"
[cycle = 162] "mshr busy"
[cycle = 162] "directory valid"
[cycle = 163] "mshr busy"
[cycle = 163] "waiting for grant ack"
[cycle = 163] "waiting for probe ack"
[cycle = 164] "mshr busy"
[cycle = 164] "schedule a probe"
[cycle = 164] "waiting for grant ack"
[cycle = 164] "waiting for probe ack"
[cycle = 165] "mshr busy"
[cycle = 165] "waiting for grant ack"
[cycle = 165] "waiting for probe ack"
```

查directory，发现Core0占有该Block

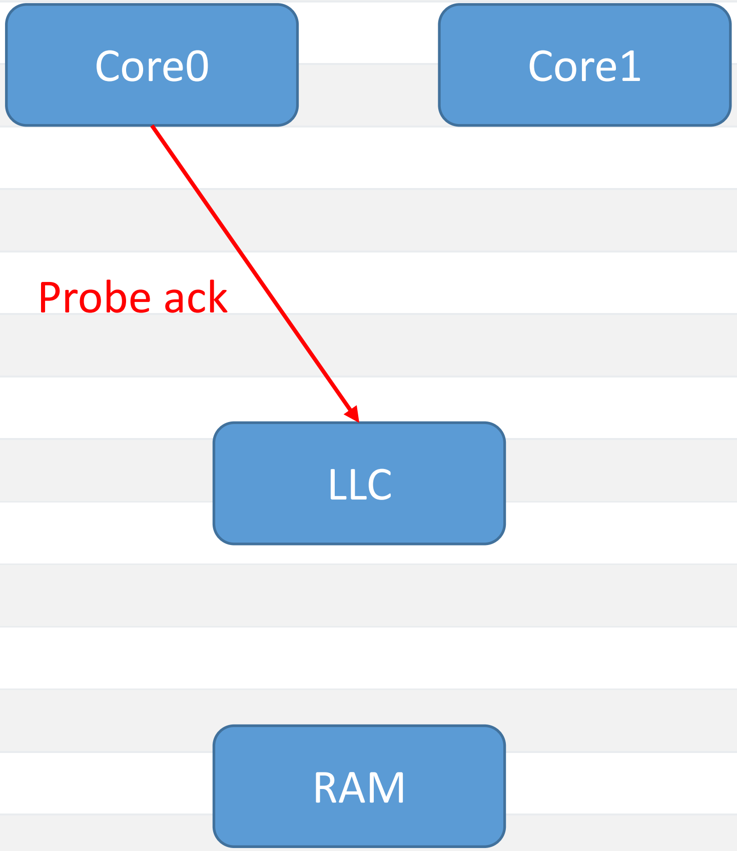
Probe Core0

Waveform Terminator: 应用示例—Log分析

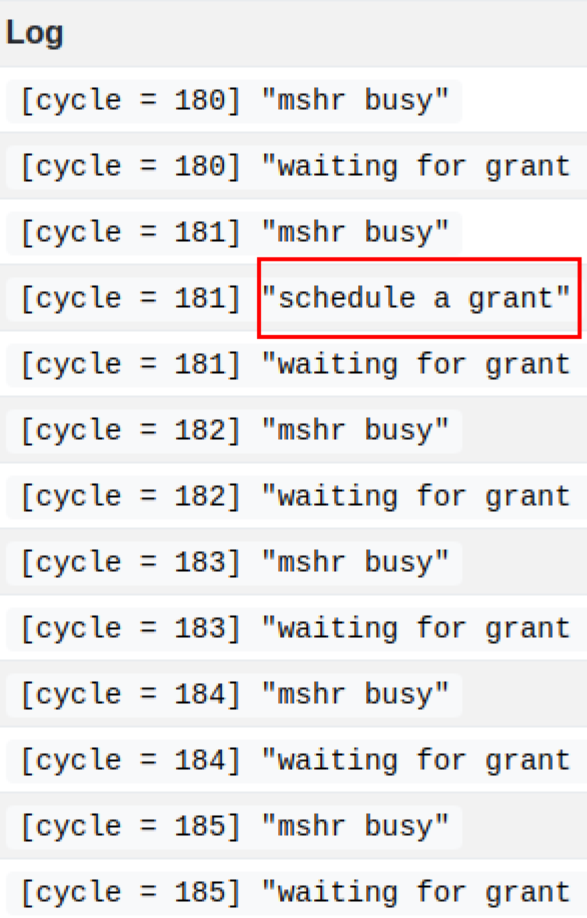


```
Log
[cycle = 175] "mshr busy"
[cycle = 175] "waiting for grant ack"
[cycle = 175] "waiting for probe ack"
[cycle = 176] "mshr busy"
[cycle = 176] "waiting for grant ack"
[cycle = 176] "waiting for probe ack"
[cycle = 177] "mshr busy"
[cycle = 177] "waiting for grant ack"
[cycle = 177] "waiting for probe ack"
[cycle = 178] "mshr busy"
[cycle = 178] "waiting for grant ack"
[cycle = 178] "waiting for probe ack"
[cycle = 179] "mshr busy"
[cycle = 179] "waiting for grant ack"
[cycle = 180] "mshr busy"
[cycle = 180] "waiting for grant ack"
```

收到probe ack

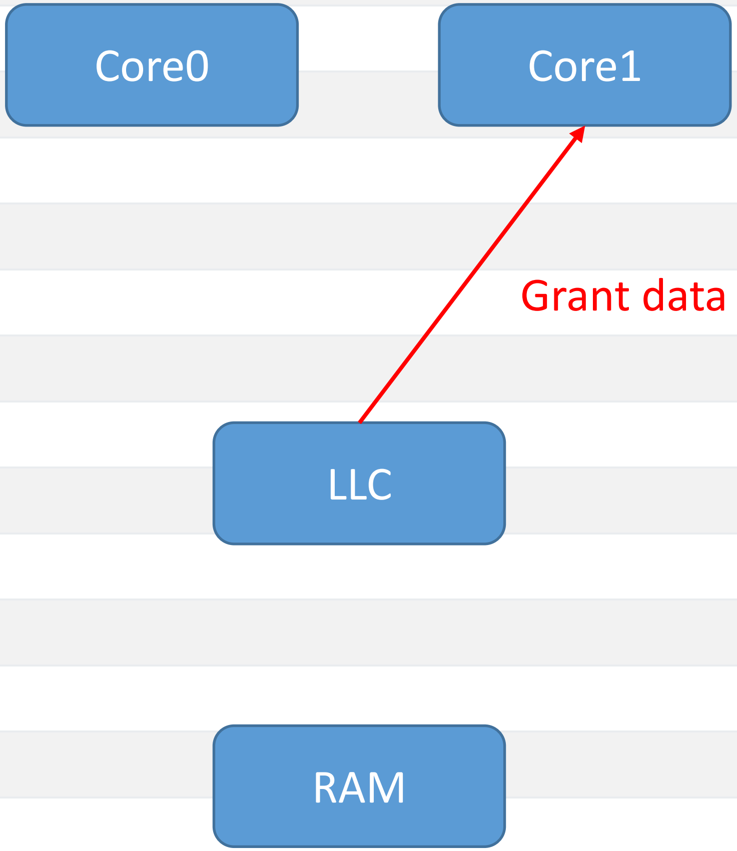


Waveform Terminator: 应用示例—Log分析

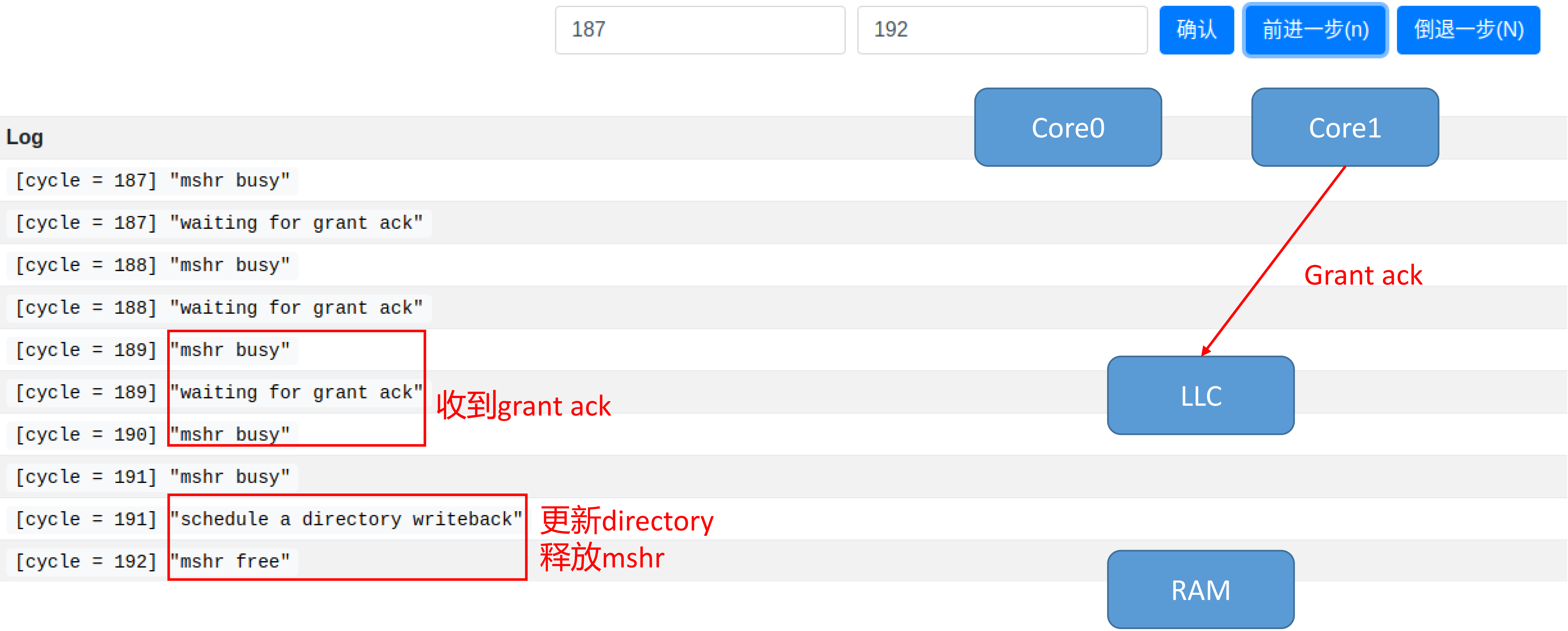


```
Log
[cycle = 180] "mshr busy"
[cycle = 180] "waiting for grant ack"
[cycle = 181] "mshr busy"
[cycle = 181] "schedule a grant"
[cycle = 181] "waiting for grant ack"
[cycle = 182] "mshr busy"
[cycle = 182] "waiting for grant ack"
[cycle = 183] "mshr busy"
[cycle = 183] "waiting for grant ack"
[cycle = 184] "mshr busy"
[cycle = 184] "waiting for grant ack"
[cycle = 185] "mshr busy"
[cycle = 185] "waiting for grant ack"
```

向Core1 grant数据



Waveform Terminator: 应用示例—Log分析



Log

```
[cycle = 187] "mshr busy"
[cycle = 187] "waiting for grant ack"
[cycle = 188] "mshr busy"
[cycle = 188] "waiting for grant ack"
[cycle = 189] "mshr busy"
[cycle = 189] "waiting for grant ack"
[cycle = 190] "mshr busy"
[cycle = 191] "mshr busy"
[cycle = 191] "schedule a directory writeback"
[cycle = 192] "mshr free"
```

收到grant ack

更新directory
释放mshr