COD21-GRP67 验收展示

罗富文 刘家宏 王拓为

基础功能

- 运行监控程序基础版本: kernel_basic.bin
- R：显示 31 个用户程序寄存器值
- A：输入 01 02 03 04，放置于地址 0x 80 40 00 00
- D：显示从地址 0x 80 40 00 00 开始，4 字节长度数据
- G：执行 5 个指定测试程序
  - <UTEST_SIMPLE>: 0x80001000
  - <UTEST_1PTB>: 0x80001008
  - <UTEST_2DCT>: 0x80001024
  - <UTEST_3CCT>: 0x80001064
  - <UTEST_4MDCT>: 0x80001080
  - <UTEST_CRYPTONIGHT>: 0x800010a8

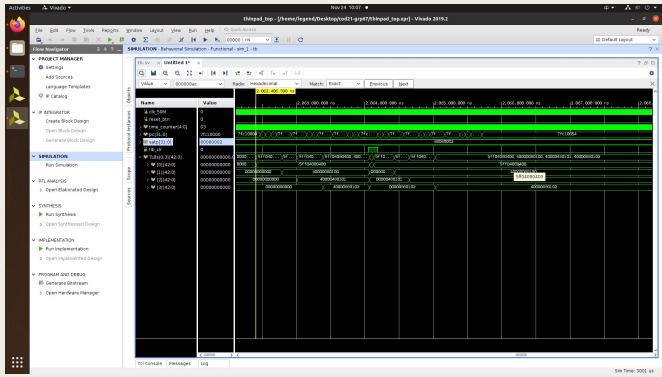
扩展功能(1)：中断异常

- 运行监控程序扩展版本: kernel_int_paging.bin
- F：从文件读入汇编指令放置于地址 0x 80 10 00 00
- G：执行指定测试程序
  - <11_ebreak>: 执行 ebreak 指令，程序立即返回
  - <12_ecall>: 执行 ecall 指令，输出 Ucore is running... 后返回
  - <13_timeout>: 执行死循环，触发程序时钟中断
  - <14_ecall_timeout>:执行 ecall 指令，输出 Ucore is running... 后执行死循环，触发程序时钟中断

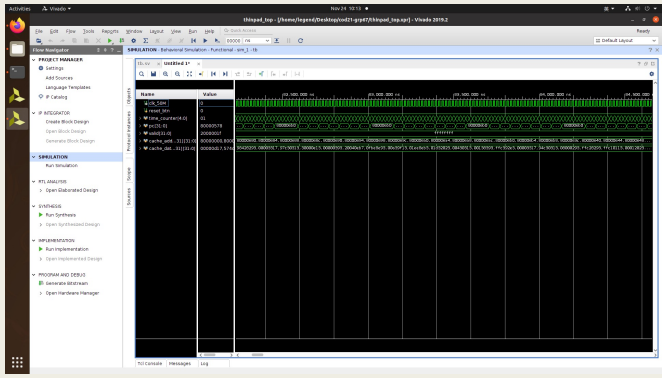
扩展功能(2)：页表支持

- 运行监控程序扩展版本: kernel_int_paging.bin
- F：从文件读入<21_paging>放置于地址 0x 80 10 00 00
- G：从地址 0x 00 00 00 00 执行指定测试程序
- F：从文件读入<21_ paging>放置于地址 0x 80 3F 00 00
- G：从地址 0x 00 2F 00 00 执行指定测试程序
- F：从文件读入<21_ paging>放置于地址 0x 80 40 00 00
- G：从地址 0x 7F C1 00 00 执行指定测试程序
- F：从文件读入<21_ paging>放置于地址 0x 80 7E 00 00
- G：从地址 0x 7F FF 00 00 执行指定测试程序

扩展功能(3)：缓存加速

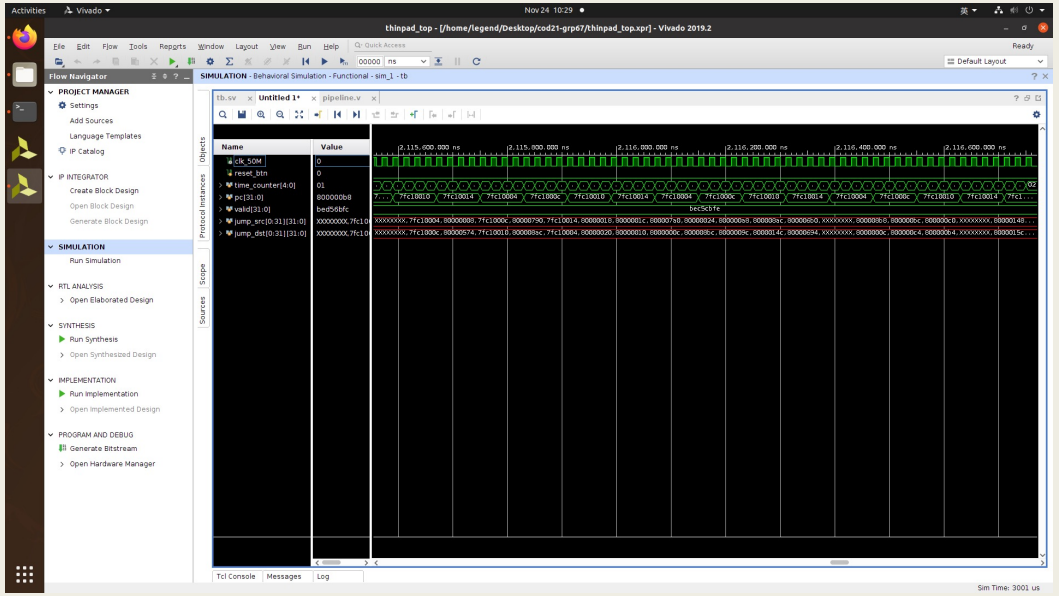


- 运行监控程序扩展版本: kernel_int_paging.bin
- 我们实现了 TLB（4项） 和 Cache（32项） 两种缓存
- 性能的提升可以通过 5 项性能测试全部通过体现
- 如有需要，可以针对以下测例进行功能仿真，查看波形图：
- 针对 TLB 的测例：<31_tlb>
  - 从虚拟地址 0x 00 00 00 00、0x 00 00 10 00 … 0x 00 00 40 00处读取数据
  - 使用 sfence.vma 指令刷新 TLB
  - 从虚拟地址 0x 00 00 00 00、0x 00 00 10 00 … 0x 00 00 40 00处读取数据
- 针对 Cache 的测例：<32_cache>
  - 从虚拟地址 0x 00 00 00 00、0x 00 00 10 00 … 0x 00 01 F0 00处写入数据
  - 从虚拟地址 0x 00 00 00 00、0x 00 00 10 00 … 0x 00 01 F0 00处读取数据



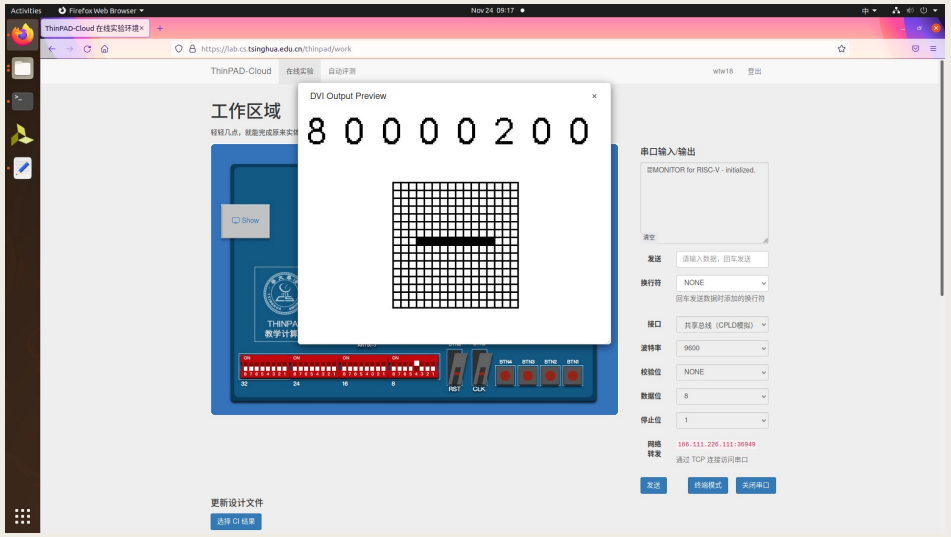
扩展功能(4)：动态预测

- 运行监控程序扩展版本: kernel_int_paging.bin
- 性能的提升可以通过 5 项性能测试全部通过体现
- 如有需要，可以针对以下测例进行功能仿真，查看波形图：
- 针对动态分支预测的测例：<41_branch_prediction>
- 执行一段 1000 次的循环



扩展功能(5)：外设驱动

- 运行监控程序扩展版本: kernel_int_paging.bin
- 我们实现了 VGA 外设
- 可以根据拨码开关显示制定寄存器的当前值以及特色应用的显示
- 注：显示值与 R 指令输出值可能不同，这是由于某个寄存器的当前值并不一定等于该寄存器在用户内存中的保存值



扩展功能(6)：特色应用

- 为了更好的体现系统设计特点，我们实现了可写入的 Conway’s Game of Life
- 游戏规则：
  - 一个状态的下一个状态完全由该状态决定。
  - 对于一个死细胞(白色)，若其周围 8 个细胞中恰好有 3 个活细胞，则其下一状态复活，否则继续保持死状态;
  - 对于一个活细胞(黑色)，若其周围 8 个细胞中有 2 个或 3 个活细胞，则其下一状态保持活状态，否则死去。
  - 一些形状的细胞会有稳定的演变周期。
- 使用说明：
  - 在运行状态下，按照转换规则每 0.5s 转换一次。
  - lock 和 trigger 同时为 1 才会使其处于运行状态，否则为停止状态。
  - trigger 初始为0，通过运行两个特殊的用户程序来转换值:
    - <61_game_trigger_start> 会将其置 1;
    - <62_game_trigger_stop> 会将其置 0;
  - 在停止状态下才可以进行写操作：
    - 在 write 为 1 时会将 x 行 y 列的细胞置活（黑色）；
    - clear 为 1 时会将 x 行 y 列的细胞置死（白色）；

扩展功能(7)：完整指令

- 运行监控程序扩展版本: kernel_int_paging.bin
- 我们实现了完整的 Rv32I 指令
- 针对扩展指令的测例：<71_extra_instr>
  - t3 = 0x 00 00 00 07
  - t5 = 0x 00 00 FF FF & t6 = 0x FF FF 00 00
  - s3 = 0x 00 00 00 2D

```
.section .text
.globl _start
_start:
    # XOR
    ori  t1, x0, 5      # t1 = 0b101
    ori  t2, x0, 2      # t2 = 0b010
    xor  t3, t1, t2     # t3 = 0x00000007
    # LH & SH
    lui  t0, 0x7fc10    # t0 = 0x7fc10000
    ori  t4, x0, -1     # t4 = 0xffffffff
    sw   t4, (t0)       # (0x7fc10000) = 0xffffffff
    lh   t5, (t0)       # t5 = 0x0000ffff
    sh   t0, (t0)       # (0x7fc10000) = 0xffff0000
    lw   t6, (t0)       # t6 = 0xffff0000
    # BLT
    ori  s1, x0, 1      # s1 = 1
    ori  s2, x0, 10     # s2 = 10
_loop:
    add  s3, s3, s1     # s3 = s3 + s1
    addi s1, s1, 1      # s1 = s1 + 1
    bne  s1, s2, _loop
_end:
    jr   ra
```

谢谢各位！