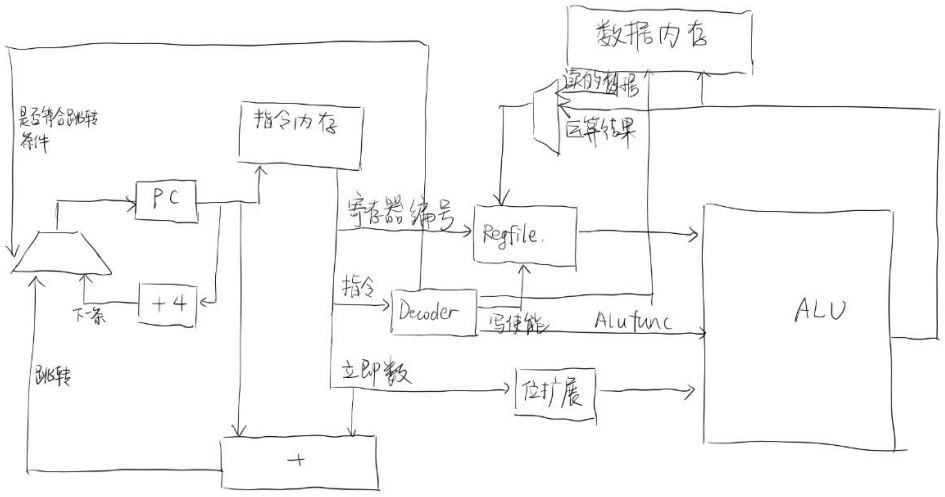
Lab2 增加访存指令并跑通测试框架

22300240028 傅文杰

2024 年 3 月 17 日

目录

1 单周期简易电路图



2 简要说明设计以及部件

电路图大部分内容和 Lab1 一致，只需要增加和内存的交互。ALU 的输出结果可能写到寄存器中，也可能写到内存中；同样，读取内存后要将结果写到寄存器中。由于有了 ibus 和 dbus，不需要自己手动写指令内存和数据内存。部件和 Lab1 一致。

3 实验中遇到的困难以及解决方法

- 在 test-lab1 中,我们需要注意读取指令内存的延迟,发起请求后,只有当收到了 iresp.data_ok 之后，才能更新 pc，并且 commit 到 difftest 中。局部的代码如下：

```
...
// pc更新
always_ff @(posedge clk) begin
    if(reset) begin
        pc <= PCINIT;
    end else if (iresp.data_ok) begin
        pc <= pc_nxt;
    end
end
...
// difftest commit接线
DifftestInstrCommit DifftestInstrCommit(
    .clock          (clk),
    .coreid         (0),
    .index          (0),
    .valid          (dreq.valid ? dresp.data_ok : iresp.data_ok),
    .pc             (pc),
    .instr          (raw_instr),
    .skip           (0),
    .isRVC          (0),
    .scFailed       (0),
    .wen            (regwrite),
    .wdest          ({3'b0, rd}),
    .wdata          (wdata)
);
...
```

- 在 test-lab2 中，面对读写内存的请求时，会有随机的延迟，一些相关变量需要保持不变，直到收到 data_ok 的信号。用时序逻辑进行跨时钟周期的保存:

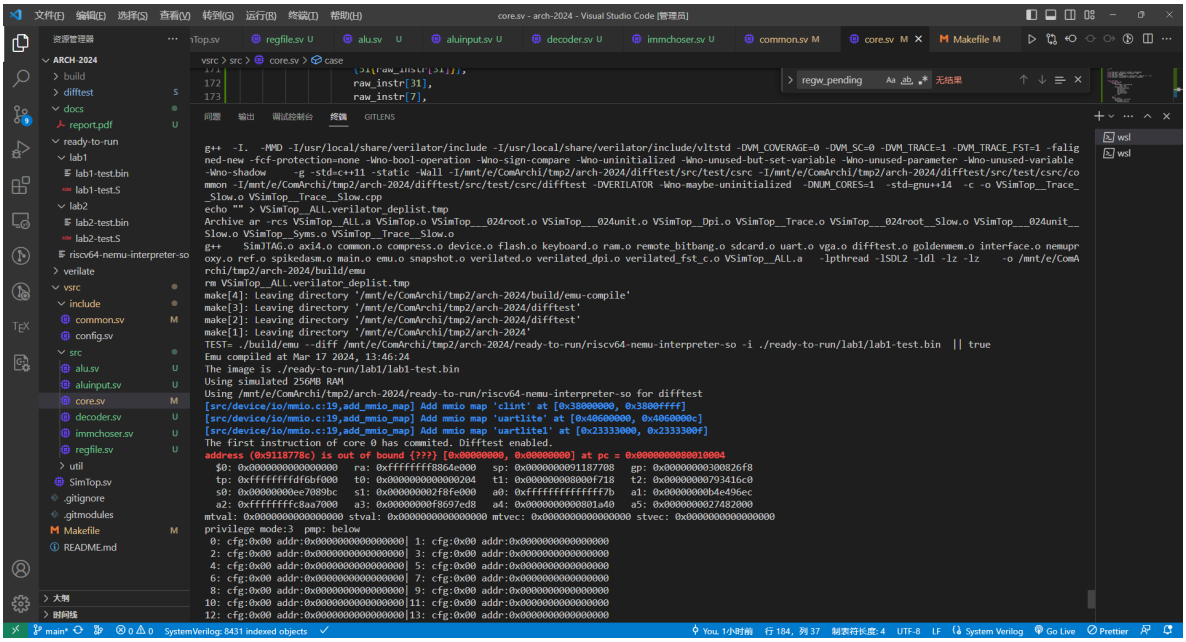
```
...
always_ff @(posedge clk) begin
    if (reset) begin
        dreq_pending <= 1'b0;
        saved_variant <= '0;
```

```
        ...
    end
    else if (dreq.valid && !dreq_pending) begin
        dreq_pending <= 1'b1;
        saved_variant <= variant;
        ...
    end
    else if (dresp.data_ok) begin
        dreq_pending <= 1'b0;
        saved_variant <= xx;
        ...
    end
end
...
```

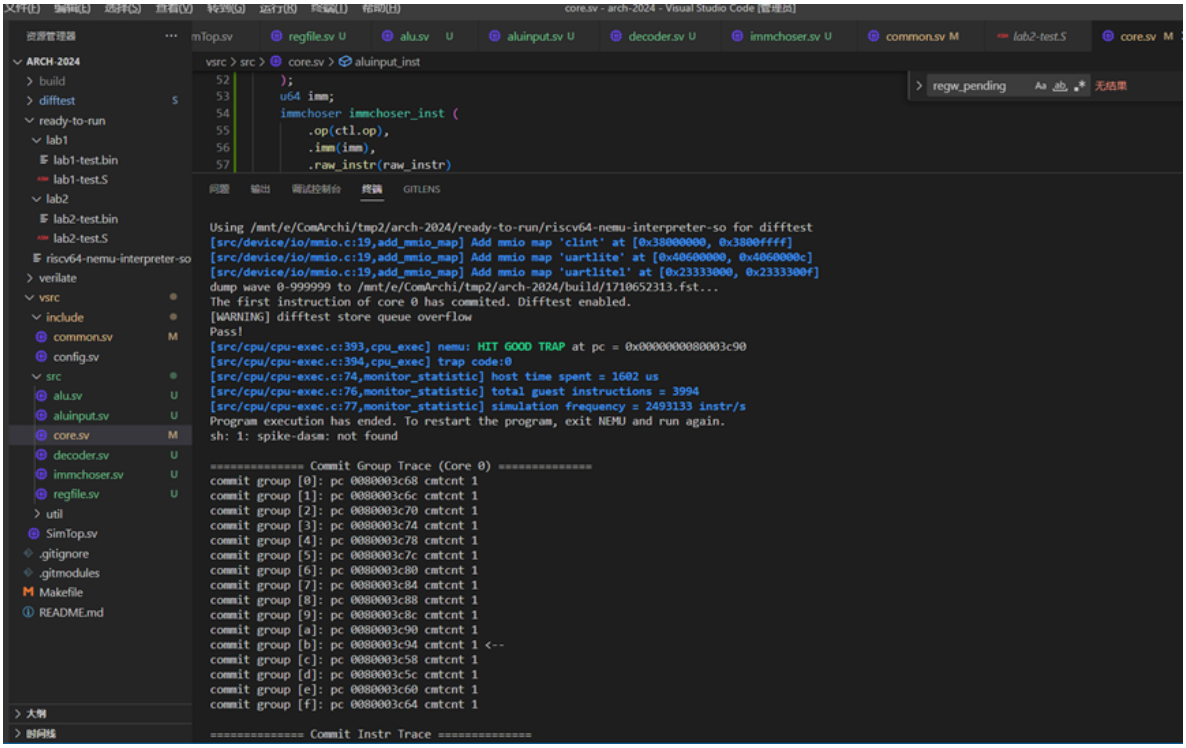
- 同样需要注意 pc 和 commit 时候的接线，将 test-lab1 中的 iresp.data_ok 改成 dreq.valid ? dresp.data_ok : iresp.data_ok 即可。

4 difftest 截图

- test-lab1



- test-lab2


core.sv - arch-2024 - Visual Studio Code [管理员]
ARCH-2024
build
difftest
ready-to-run
lab1
lab1-test.bin
lab1-test.S
lab2
lab2-test.bin
lab2-test.S
riscv64-nemu-interpreter-so
verilate
vsrc
include
common.sv
config.sv
src
alu.sv
aluinput.sv
core.sv
decoder.sv
immchoser.sv
regfile.sv
util
SimTop.sv
.gitignore
.gitmodules
Makefile
README.md
vsrc > src > core.sv > aluinput_inst
u64 imm;
immchoser immchoser_inst (
.op(ctl.op),
.imm(imm),
.raw_instr(raw_instr)
regw_pending
无结果
Using /mnt/e/ComArchi/tmp2/arch-2024/ready-to-run/riscv64-nemu-interpreter-so for difftest
[src/device/io/mmio.c:19,add_mmio_map] Add mmio map 'clint' at [0x38000000, 0x3800ffff]
[src/device/io/mmio.c:19,add_mmio_map] Add mmio map 'uartlite' at [0x40600000, 0x4060000c]
[src/device/io/mmio.c:19,add_mmio_map] Add mmio map 'uartlite1' at [0x23333000, 0x2333300f]
dump wave 0-999999 to /mnt/e/ComArchi/tmp2/arch-2024/build/1710652313.fst...
The first instruction of core 0 has commited. Difftest enabled.
[WARNING] difftest store queue overflow
Pass!
[src/cpu/cpu-exec.c:393,cpu_exec] nemu: HIT GOOD TRAP at pc = 0x0000000080003c90
[src/cpu/cpu-exec.c:394,cpu_exec] trap code:0
[src/cpu/cpu-exec.c:74,monitor_statistic] host time spent = 1602 us
[src/cpu/cpu-exec.c:76,monitor_statistic] total guest instructions = 3994
[src/cpu/cpu-exec.c:77,monitor_statistic] simulation frequency = 2493133 instr/s
Program execution has ended. To restart the program, exit NEMU and run again.
sh: 1: spike-dasm: not found
========== Commit Group Trace (Core 0) ==========
commit group [0]: pc 0080003c68 cmtcnt 1
commit group [1]: pc 0080003c6c cmtcnt 1
commit group [2]: pc 0080003c70 cmtcnt 1
commit group [3]: pc 0080003c74 cmtcnt 1
commit group [4]: pc 0080003c78 cmtcnt 1
commit group [5]: pc 0080003c7c cmtcnt 1
commit group [6]: pc 0080003c80 cmtcnt 1
commit group [7]: pc 0080003c84 cmtcnt 1
commit group [8]: pc 0080003c88 cmtcnt 1
commit group [9]: pc 0080003c8c cmtcnt 1
commit group [a]: pc 0080003c90 cmtcnt 1
commit group [b]: pc 0080003c94 cmtcnt 1 <--
commit group [c]: pc 0080003c58 cmtcnt 1
commit group [d]: pc 0080003c5c cmtcnt 1
commit group [e]: pc 0080003c60 cmtcnt 1
commit group [f]: pc 0080003c64 cmtcnt 1
========== Commit Instr Trace ==========