

实验：单周期 CPU lab2

实验人：谢志康

学号：22307110187

日期：2024.3.19

代码架构建立在 lab1 的基础上，仅新增了 Fetch 模块，并且修改了部分 Control.sv 和 MEM.sv 内容，其它部分基本不变，并将整个代码框架与测试环境相接来测试 CPU 运行状态。每个模块的接口功能以及 CPU 总流程图均与 lab1 所交实验报告中介绍相同。

解决内存访问冲突的总思路：利用 stall, stall_next_ibus, stall_this_dbus 三个使能来控制 PC。

stall_next_ibus: 当前指令执行阶段，判断下一条指令是否已经从内存中读取出来。

stall_this_dbus: 当前指令执行阶段，判断访存操作是否已经完成。

stall = stall_next_ibus || stall_this_dbus

由于访存操作（主要是 MEM、FETCH 中涉及）需要多个周期，因此依靠这两个 stall 作为使能来控制 PC，只要有一个还没结束就把 PC 按在原地空转，没法把下一条指令取出来。

简述每个具有 stall 的模块的执行逻辑——

Fetch.sv: 取指，并依靠这个模块更新 ibus 总线的 iresp、ireq 信息。

NPC 模块计算完后给出的下个 pc 值还是直接赋给 ireq 地址就行，若被 stall，PC 会空转，NPC 不会继续往前更新。

若当前指令有访存操作，且未完成，则 stall_this_dbus 会被设为 1，ireq.valid 应置为无效，vice versa。（每个 CPU 周期从取指开始，把指令取出来之后经过一系列组合逻辑去译码，译码后如果发现访存操作的话就需要把取指的 ireq.valid 先变成 0，按住不让直接取下一条指令，然后等访存结束之后重新把 ireq.valid 置成 1，等待下一条指令被取出，开始下一个 CPU 周期）

指令响应中，若 data 或 addr 任意一个没有准备好，都不应该访存取出下一个指令，否则会导致冲突，依靠 stall_next_ibus 来阻塞，将其置 1，不让取下一条指令。

PC.sv: 当 CPU 运行时若 stall 为 1，则表示之前的操作还没有执行完，这个时候 PC 不能再往后更新了，则在 PC 模块中设置让 PC 空转一段时间 ($PC \leq PC$)，直到 stall 为 0，也即之前的操作全部做完，并且在时钟上升沿时，再将 PC 更新为 NPC 模块给出的下一个值。

（补充说一下：lab1 交上去的代码 sext 中 u 型指令扩展错了不好意思……已经改正）

RegFile.sv: 传入 stall 信号，起阻塞作用，若 stall 为 1，则不对 Reg 执行写入操作，保持 regfile 的值不改变。

MEM.sv: 总线 dreq 的 data 和 addr 直接写入更新即可。若当前 CPU 周期解析的指令有访存操作，则在 control 模块中 dbus_wre 会被置为 1 表示读写内存使能，同时，当这个访存操作未完成时，dreq.valid 持续拉高，stall_this_dbus 被置为 1，表示当前指令执行阶段访存操作还未完成，CPU 应该空转一段时间等它完成。

Test-lab1 运行截图:

```
Windows PowerShell x kurumi@kurumi: ~/arch-2024 x + v
fs4: 0x6781213739a1c137 fs5: 0xbe3a50b76de89337 fs6: 0xc789f1371be37193 fs7: 0x42014313001141b3
fs8: 0x081532b7e5a16113 fs9: 0x3771421300334133 fs10: 0x341df197dd0c3137 fs11: 0x4a9102370091e133
ft8: 0x9ac72497003081b3 ft9: 0x80837197efd3f0b7 ft10: 0x0000019785a82137 ft11: 0x0e818093ee8180e7
pc: 0x0000000000010002 mstatus: 0x0000000000000000 mcause: 0x0000000000000000 mepc: 0x0000000000000000
ssstatus: 0x0000000000000000 scause: 0x0000000000000000 sepc: 0x0000000000000000
satp: 0x0000000000000000
mip: 0x0000000000000000 mie: 0x0000000000000000 mscratch: 0x0000000000000000 sscratch: 0x0000000000000000
mideleg: 0x0000000000000000 medeleg: 0x0000000000000000
mtval: 0x0000000000000000 stval: 0x0000000000000000 mtvec: 0x0000000000000000 stvec: 0x0000000000000000
privilege mode: 3 pmp: below
0: cfg:0x00 addr:0x0000000000000000 1: cfg:0x00 addr:0x0000000000000000
2: cfg:0x00 addr:0x0000000000000000 3: cfg:0x00 addr:0x0000000000000000
4: cfg:0x00 addr:0x0000000000000000 5: cfg:0x00 addr:0x0000000000000000
6: cfg:0x00 addr:0x0000000000000000 7: cfg:0x00 addr:0x0000000000000000
8: cfg:0x00 addr:0x0000000000000000 9: cfg:0x00 addr:0x0000000000000000
10: cfg:0x00 addr:0x0000000000000000 11: cfg:0x00 addr:0x0000000000000000
12: cfg:0x00 addr:0x0000000000000000 13: cfg:0x00 addr:0x0000000000000000
14: cfg:0x00 addr:0x0000000000000000 15: cfg:0x00 addr:0x0000000000000000
privilegeMode: 3
this_pc_differnt at pc = 0x0000010002, right= 0x0000000000010002, wrong = 0x0000000000010004
Core 0 ABORT at pc = 0x00010004
total guest instructions = 0
instrCnt = 0, cycleCnt = 0, IPC = -nan
Seed=0 Guest cycle spent: 1972229 (this will be different from cycleCnt if emu loads a snapshot)
Host time spent: 1062ms
This emulator compiled with JTAG Remote Bitbang client. To enable, use +jtag_rbb_enable=1.
Listening on port 23334
kurumi@kurumi:~/arch-2024/arch-2024$
```

Test-lab2 运行截图:

```
Windows PowerShell x kurumi@kurumi: ~/arch-2024 x + v
[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]
The first instruction of core 0 has committed. Difftest enabled.
[WARNING] difftest store queue overflow
Pass!
[src/cpu/cpu-exec.c:393,cpu_exec] nemu: HIT GOOD TRAP at pc = 0x0000000000003c90
[src/cpu/cpu-exec.c:394,cpu_exec] trap_code:0
[src/cpu/cpu-exec.c:74,monitor_statistic] host time spent = 1827 us
[src/cpu/cpu-exec.c:76,monitor_statistic] total guest instructions = 3994
[src/cpu/cpu-exec.c:77,monitor_statistic] simulation frequency = 2186097 instr/s
Program execution has ended. To restart the program, exit NEMU and run again.
sh: 1: spike-dasm: Permission denied

===== 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
```