

中国科学院大学

《计算机体系结构基础(研讨课)》实验报告

姓名 艾华春,李霄宇,王敬华

学号 2022K8009916011,2022K8009929029,2022K8009925009

实验项目编号 6 实验名称 存储管理单元设计

一、 逻辑电路结构与仿真波形的截图及说明

- TLB 模块设计。
- 添加 TLB 相关指令和 CSR 寄存器。

1. 添加 TLB 相关指令和 CSR 寄存器
- 2.

- 添加 TLB 相关例外支持,添加虚实地址映射功能。

1. 为 cpu 增加虚实映射功能

在 if 和 ex 级发出访存请求的虚拟地址,同时送往直接地址翻译,直接窗口映射和 tlb 地址映射翻译三处,同时进行各自的虚实地址转换,然后再根据 CSR 寄存器中的值进行选择。

以 if 级发出指令访存信号为例,通过组合逻辑,在同一个 clk 内生成三种地址翻译的结果,最后通过多路选择器进行选择。

访存物理地址 pre_pc_pa 首先根据控制寄存器中的 crmd.pg 中的值选择当前为直接地址地址翻译模式(物理地址 = 虚拟地址 = pre_pc),还是映射地址翻译模式(物理地址 = pre_pc_map)。

映射地址翻译(pre_pc_map)首先判断是否命中直接映射窗口,若不是,则使用 tlb 地址映射结构。

第一行将虚拟地址 pre_pc [31:12] 发送到 tlb 进行 tlb 地址翻译,tlb 中的组合逻辑查找结果 s0_ps, s0_ppn, s0_found

```
assign {s0_vppn, s0_va_bit12} = pre_pc[31:12]; // output to tlb

assign hit_dmw0 =      csr_dmw0_plv_met & csr_dmw0_vseg == pre_pc[31:29];
assign hit_dmw1 =      csr_dmw1_plv_met & csr_dmw1_vseg == pre_pc[31:29];

assign pre_pc_pa =      csr_crmd_pg ? pre_pc_map // enable mapping
:pre_pc;                // direct translate

assign pre_pc_map = hit_dmw0 ? {csr_dmw0_pseg, pre_pc[28:0]} // direct map windows 0
:hit_dmw1? {csr_dmw1_pseg, pre_pc[28:0]} // direct map windows 1
:(s0_ps == 6'b010101) ? {s0_ppn[19:9], pre_pc[20:0]} // tlb map: ps 4Mb
:{s0_ppn,pre_pc[11:0]};                               // tlb map : ps 4kb
```

2. 为 cpu 添加 tlb 相关例外支持

在每个流水级中设置 `excep_en` , `excep_ecode`, `excep_esubcode` 分别表示当前流水级是否触发异常, 和异常的种类。

以在 `ex` 流水级中为例:

```
assign ex_excep_en =    ex_excep_ALE | ex_excep_TLBR | ex_excep_PIL | ex_excep_PIS |
    ex_excep_PPI | ex_excep_PME | id_excep_en;

assign ex_excep_TLBR =    //TLB 重填例外
    (ex_res_from_mem | ex_mem_we) & csr_crmd_pg & ~hit_dmw0 & ~hit_dmw1 & ~s1_found;
assign ex_excep_PIL =    // load 操作页无效例外
    (ex_res_from_mem) & csr_crmd_pg & ~hit_dmw0 & ~hit_dmw1 & s1_found & ~s1_v;
assign ex_excep_PIS =    // store 操作页无效例外
    (ex_mem_we) & csr_crmd_pg & ~hit_dmw0 & ~hit_dmw1 & s1_found & ~s1_v;
assign ex_excep_PPI =    // 页特权等级不合规例外
    (ex_res_from_mem | ex_mem_we) & csr_crmd_pg & ~hit_dmw0 & ~hit_dmw1 & s1_found & s1_v &
    (s1_plv < csr_crmd_plv);
assign ex_excep_PME =    // 页修改例外
    (ex_res_from_mem | ex_mem_we) & csr_crmd_pg & ~hit_dmw0 & ~hit_dmw1 & s1_found & s1_v &
    (s1_plv >= csr_crmd_plv) & ~s1_d;

assign ex_badv =        (id_excep_en) ? id_badv
    : ex_alu_result;
assign ex_esubcode =    (id_excep_en) ? id_esubcode
    : 9'b0;
assign ex_ecode =        (id_excep_en) ? id_ecode
    : ex_excep_ALE ? 6'h9
    : ex_excep_TLBR ? 6'h3f // tlb refill
    : ex_excep_PIL ? 6'h1
    : ex_excep_PIS ? 6'h2
    : ex_excep_PPI ? 6'h7
    : 6'h4; // pme
```

所有 `tlb` 相关的异常, 触发的必要条件都有: 当前指令是访存指令 (`load/store`), 当前虚拟地址翻译模式不是直接地址翻译模式, 且不命中直接映射窗口。

即 $(\text{ex_res_from_mem} \mid \text{ex_mem_we}) \& \text{csr_crmd_pg} \& \sim \text{hit_dmw0} \& \sim \text{hit_dmw1}$

(a) TLB 重填例外

在 `tlb` 没有找到对应的页表项, 即 `s1_found == 0`

(b) `load` 操作页无效例外, `store` 操作页无效例外

在 `tlb` 中找到对应的页表项, 但是页表项的有效位为 0, 即 `s1_found & ~s1_v`

(c) 页特权等级不合

在 `tlb` 中找到对应的页表项, 页表项的有效, 但当前特权等级为 `plv3`, 但是页表项所需要特权等级为 `plv0`, 即 `s1_found & s1_v & (s1_plv < csr_crmd_plv)`

(d) 页修改例外在 `tlb` 中找到对应的页表项, 页表项的有效, 且特权等级满足要求, 但是页表项脏位为 0, 即 `s1_found & s1_v & (s1_plv >= csr_crmd_plv) & ~s1_d`

由上面的分析可知, 与 `tlb` 相关的异常触发条件互斥, 不用考虑不同异常间的优先级的的问题。

二、 实验过程中遇到的问题、对问题的思考过程及解决方法(比如 RTL 代码中出现的逻辑 bug, 逻辑仿真和 FPGA 调试过程中的难点等)

- if 级错误丢弃异常处理函数的第一条指令。

要求在发生 mmu 相关异常的时候, 不能向总线发出请求, 所以在 pre_if 级中识别到 pc 的虚拟地址发生异常, 则将 inst_sram_req 拉低。

但是导致了异常传到 wb 级时, 发出更新流水线信号 flush 时, 将 if 级误判为还在等待指令返回(实际上 pre_if 级没有发出请求), 则将 inst_cancel 信号(表示要丢弃接收到的第一条指令)拉高, 导致丢弃了异常处理函数的第一条指令。

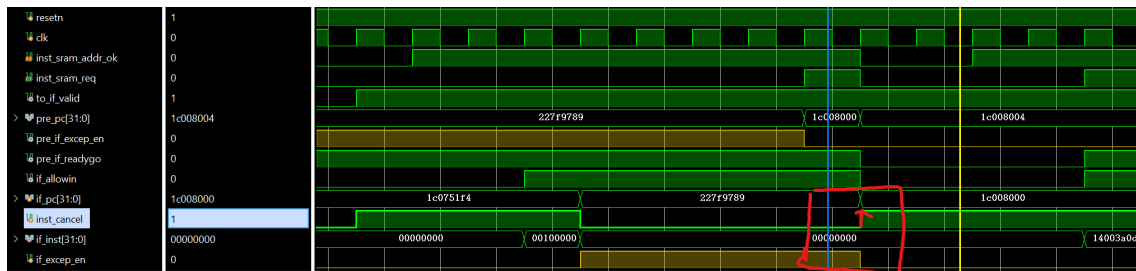


图 1: 错误拉高 inst_cancel

修改: 在 inst_cancel 拉高的逻辑中, 判断 if 级是否有异常, 若有, 则说明 if 级没有已经发出请求, 但是没有接受到的总线事务, 不用拉高 inst_cancel。

```
/* 清空流水线时, 第一个指令需要丢弃*/
always @(posedge clk) begin
    if(~resetn)
        inst_cancel <= 1'b0;
    else if ( (if_valid & ~if_ir_valid & ~inst_sram_data_ok & ~if_except_en // if正在等待指令返回,
        加入if级是否有异常的判断
        |pre_if_reqed_reg & ~inst_sram_data_ok) // pre_if
        级发出请求, 但是数据没有返回, 也还没有进入if级
        & (flush | br_taken))
        inst_cancel <= 1'b1;
    else if(inst_sram_data_ok) // 异常后第一个需要被舍弃的指令返回
        inst_cancel <= 1'b0;
end
```

三、 小组成员分工合作情况

王敬华负责 ex 级访存添加类 sram 总线支持。

李霄宇负责部分 debug 工作。

艾华春负责添加 TLB 相关寄存器和 TLB 相关的例外支持, 为 cpu 增加虚实映射功能。

实验报告为根据每人负责代码的部分, 写相应部分的报告。