浙江大学

本科实验报告

| 课程名称: | 计算机体系结构 | | |
|-------|------------|--|--|
| 姓 名: | 汪辉 | | |
| 学 院: | 计算机科学与技术学院 | | |
| 系: | | | |
| 专业: | 计算机科学与技术 | | |
| 学 号: | 3190105609 | | |
| 指导教师: | 陈文智 | | |

2021年 11月 9日

浙江大学实验报告

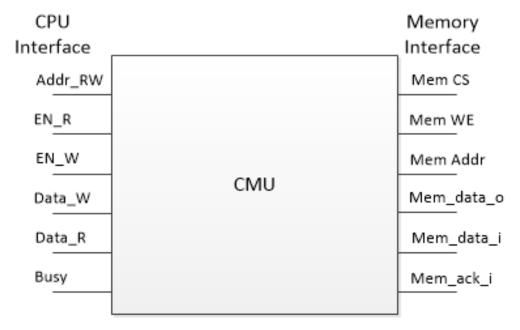
| 课程名称: | 计算机体 | 系结构 | 实验 | 类型: _ | 综合 | 1 |
|------------------|-----------|----------------|-----------------|----------|-----|---|
| 实验项目名称: | Pipelined | CPU with Cache | | | | |
| 学生姓名: <u>汪</u> 耀 | <u> </u> | 机科学与技术: | 学号: <u>319</u> | 00105609 | 9_ | |
| 同组学生姓名: | 王嘉豪 | | ≨导老师 : _ | 防 | 下文智 | |
| 实验地点: | 曹西 301 | 实验日期: | 2021 | 年 11 | 月 9 | 日 |

一、实验目的和要求

- Understand the principle of Cache Management Unit (CMU) and State Machine of CMU
- Master the design methods of CMU and Integrate it to the CPU.
- Master verification methods of CMU and compare the performance of CPU when it has cache or not.

二、实验内容和原理

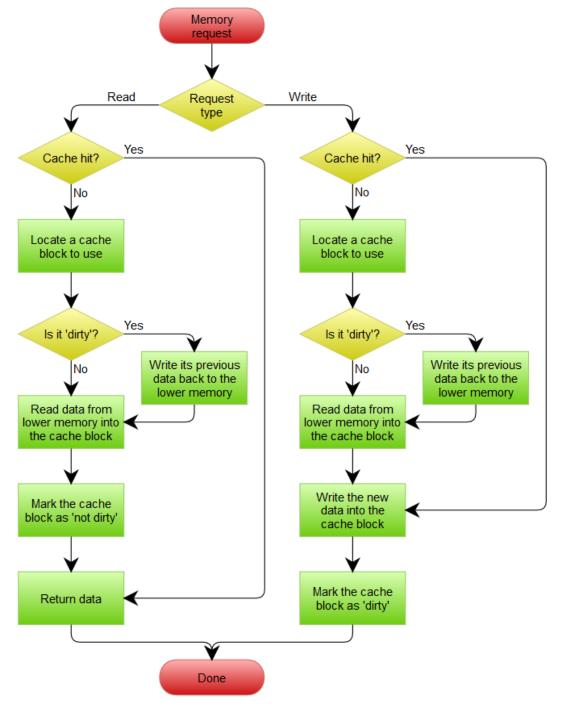
- Design of Cache Management Unit and integrate it to CPU.
- Observe and Analyze the Waveform of Simulation.
- Compare the performance of CPU when it has cache or not.



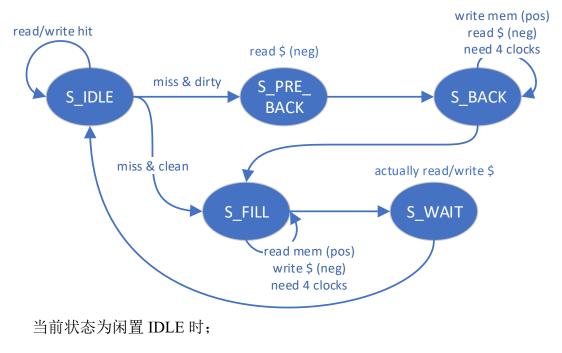
三、 实验过程和数据记录

完善 Cache Management Unit 模块,CMU 模块主要围绕三个问题设计,hit、miss 和 write_back,写内存时不立即写回 memory 而是先写到 cache,直到 cache 中的块被替换时才写回 memory,即 write back 策略。

参考指导中的流程设计状态机来完成整个模块:



实验工程中已经设计好状态机的框架,需要完善的只有各个状态之间的切换和一些信号:



```
S_IDLE: begin
      if (en_r || en_w) begin
          if (cache_hit)
              next_state = S_IDLE ; // need filled
          else if (cache_valid && cache_dirty)
              next_state = S_PRE_BACK ; // need filled
          else
              next_state = S_FILL ;
      end
      next_word_count = 2'b00;
  end
PRE BACK:
 S_PRE_BACK: begin
     next_state = S_BACK ; // need filled
     next_word_count = 2'b00;
 end
BACK:
```

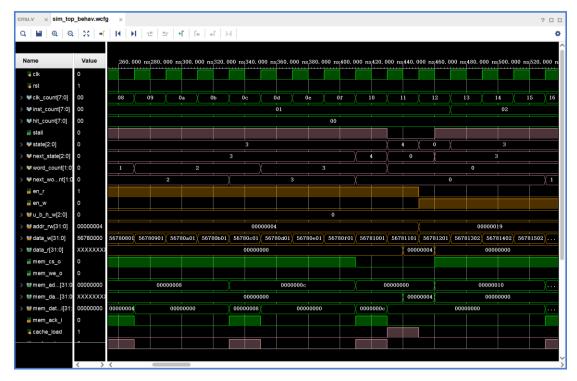
```
S_BACK: begin
   if (mem_ack_i && word_count == {ELEMENT_WORDS_WIDTH{1'b1}}) // 2'b11 in default case
      next_state = S_FILL ; // need filled
     next_state = S_BACK ; // need filled
   if (mem_ack_i)
      next_word_count = word_count+1 ; // need filled
      next_word_count = word_count;
end
 FILL:
S FILL: begin
    if (mem_ack_i && word_count == {ELEMENT_WORDS_WIDTH{1'b1}})
         next_state = S_WAIT ; // need filled
    else
         next_state = S_FILL ; // need filled
    if (mem_ack_i)
         next_word_count = word_count+1 ; // need filled
    else
         next_word_count = word_count;
end
 WAIT:
  S_WAIT: begin
       next state = S IDLE : // need filled
       next_word_count = 2'b00;
  end
```

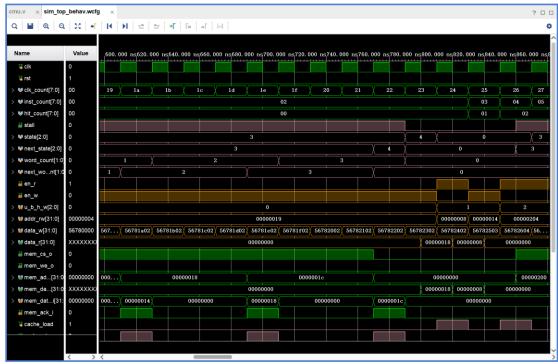
IDLE 状态代表直接命中的情形,此时不需要多余的周期来访问 Memory 取内存,在其他状态(miss)下,内存访问的指令需要一些另外的周期来完成,此时需要插入停顿终止流水线的运行:

```
assign stall = ( next_state != S_IDLE ) ;
```

四、实验结果分析

仿真结果:





实验箱结果:

第一条访问内存指令,由于 cache 的结果为 miss,使流水线停顿 16 个周期,随后同内存地址的访问结果是 hit 不需要停顿继续运行即可。遇到 cache 为 miss

的指令又停顿:

```
Zhejiang University Computer Organization Experimental
                                  SOC Test Environment (With RISC-U)
×0:zero 00000000
                               ×01: ra 00000000
×05: t0 00000000
                                                            x02: sp 00000000
x06: t1 00000000
×04: tp 00000000
×8:fps0 00000000
                                                                                       x83: gp 88888888
x87: t2 8888888
x11: a1 88888888
x15: a5 8888888
x19: s3 88888888
x23: s7 88888888
x23: s7 88888888
                               ×09: s1 00000000
×12: a2 00000000
×16: a6 00000000
                                                            ×10: a0 00000000
                                ×13: a3 00000000
                                                            ×14: a4 00000000
×18: s2 00000000
×22: s6 00000000
                               ×17: a7 00000000
×21: s5 00000000
×20: s4 00000000
×24: s8 00000000
                               x25: s9 00000000
x29: t4 00000000
                                                             x26:s10 000000000
                                                                                       x27:s11 00000000
×28: t3 00000000
                                                             x30: t5 00000000
                                                                                       x31: t6 000000000
                                MST-IF 81090203
INST-IF 91090203
INST-ID 910902183
INST-ID 9109193
INST-ID 91090903
INST-ID 91090903
INST-ID 91090903
PC---ID 000000C
PC -MDM 88888884
PC-WB 00000000
ALU-Ain 00000000
ALU-Bin 0000001C
                                                                                        CODE-03 01C02183
                                 lw x04,x00,910H
lw x03,x00,010H
CODE-00 00000013
                                                                                        CODE-07 00000000
CODE-04 01C04203
CODE-08 00000000
                                                                                        CODE-0B 00000000
                                              lw x92, x90, 81CH
                                                                                        CODE-0F 00000000
                                                     lw x91, x90,010H
                                                             nop JStall:addi8
CODE-16 80088888
                                                                                        CODE-13 00000000
 CODE-0C 00000000
                                                                                        CODE-17 00000000
 CODE-10 00000000
                                 CODE-15 80060000
                                                                                        CODE-1B 00000000
                                                              CODE-1A 00000000
 CODE-14 99999999
                                 CODE-19 00000000
                                                                                        CODE-1F 00000000
                                                              CODE-1E 00000000
 CODE-18 00000000
                                 CODE-1D 80000000
                                                              CODE-22 00900000
                                                                                        CODE-27 00000000
                                 CODE-21 00000000
 CODE-1C 00000000
                                                              CODE-26 00000000
 CODE-20 00000000
                                 CODE-25 89898889
 CODE-24 00000000
```



由于 write_back 策略,遇到访问之前已经写过的内存时,需要额外停顿的周期来写回内存后继续访问:



五、 讨论与心得

本次实验总体较为简单,完成得也较为顺利,状态机的总体设计已经完成, 只需按照指导完善状态之前的切换即可。