# 浙江水学

# 本科实验报告

课程名称:	计算机体系结构
姓 名:	
学 院:	计算机科学与技术学院
系:	计算机科学与技术系
专业:	计算机科学与技术
学 号:	
指导教师:	卜凯

2022年 11月 23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
体性石物:	11 异饥净余辐构	<b>头</b> 狍矢空 <b>:</b>	<b>练</b> 盲

实验项目名称: Lab4: Pipelined CPU with Cache

学生姓名: 专业: 计算机科学与技术 学号:

同组学生姓名: 指导老师: 卜凯

实验地点: <u>曹西 301</u> 实验日期: <u>2022</u>年 <u>11</u>月 <u>23</u>日

# 1 Tasks and requirements

#### 1.1 Tasks

The main tasks of Lab-4 are:

- 1. Get knowledge of the principles of cache controller.
- 2. Complete the code of cache controller.

## 1.2Requirements

This experiment would be based on SWORD development board, with xc7k325tffg676-2L FPGA.

We are given a Verilog project complementing a CPU core and other components for debugging on board. Most of the parts have been finished. There is only one file we need to complete, that is:

**cmu.v**, the control unit of a 2-way set associative LRU replaced write-back cache which we finished in lab-3.

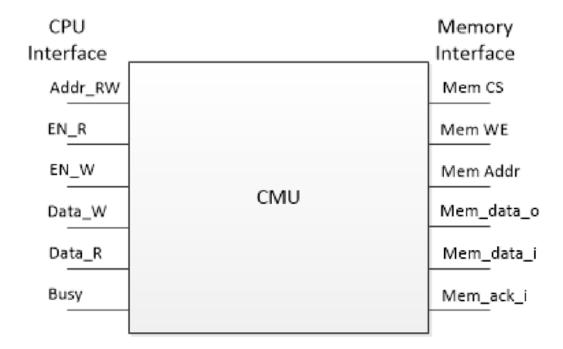
When this work is finished, we shall verify both the simulation results and the on-board results of a preset program.

# 2 Contents and principles

All of the following principles are based on RISC-V 5-stage pipelined CPU.

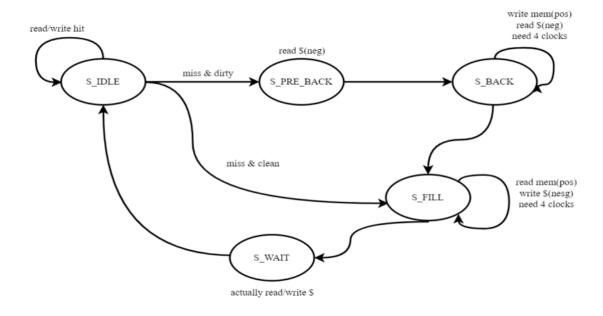
## 2.1 Cache Controller

Cache Controller is the unit for CPU-Cache communication and control. The cache controller we need to implement has following inputs and outputs, as a special instance of CPU-Memory communication:



In fact, cache controller is used to control the **load**, **edit** or **store** operations of cache. How to arrange these operations will be explained later.

### 2.2Cache Automaton



To deal with the **hit**, **dirty miss** or **clean miss** conditions, we introduced the shown automaton. The automaton has 5 determined states: **S\_IDLE**, **S\_PRE\_BACK**, **S\_BACK**, **S\_FILL**, **S\_WAIT**.

**S\_IDLE** is used to indicate that the cache is idle and does not need do anything else. If the CPU does not need to access the cache or the cache is experiencing a **hit**, the state maintains. If encountering a dirty miss, then it transfers to **S\_PRE\_BACK**; If a clean miss, then it transfers to **S\_FILL**.

**S\_PRE\_BACK** is for the preparation of **S\_BACK**. It sets the word count to 0 and load the first word from the cache for writing back to memory. It must transfer to **S\_BACK** to finish following writes.

**S\_BACK** would do the following 4 writes to the lower memory. It increases the word count by 1 at each clock cycle, write the word back to memory, and read another word from cache. When the operations are finished after 4 cycles, it will transfer to **S\_FILL**, otherwise it will maintain.

**S\_FILL** loads 4 words from memory. It increases the word count by 1 at each clock cycle, read that word from memory, and write this word to cache. When the operations are finished after 4 cycles, it will transfer to **S\_WAIT**, otherwise it will maintain.

**S\_WAIT** deals with the real query about the cache that produced that miss. At **S\_WAIT** we can guarantee it produces a hit according to the automaton. Then the whole process after miss is finished, and the state will transfer to **S\_IDLE**.

# 3 Steps and data records

## 3.1 Completed Verilog source files

#### 3.1.1 cmu.v

For clarity, we will omit some pre-set code segments, and only analysis the code blocks that we need to fill in.

The most important parts we need to finish is the parts that concerns the automaton, in the always@\* block.

In fact, the code filled here is quite simple and we need not to explain much. The

principles we need can be found in part 2.2. Following is the automaton implement:

```
always @ (*) begin
 if (rst) begin
    next\_state = S\_IDLE;
    next_word_count = 2'boo;
  end
  else begin
  case (state)
  S_IDLE: begin
    if(en_r || en_w) begin
      if (cache_hit)
        next_state = S_IDLE;
      else if (cache_valid && cache_dirty)
        next\_state = S\_PRE\_BACK;
      else
        next\_state = S\_FILL;
    end
      next_word_count = 2'boo;
    end
  S_PRE_BACK: begin
    next state = S BACK;
    next_word_count = 2'boo;
  end
 S_BACK: begin
    if (mem_ack_i && word_count == {ELEMENT_WORDS_WIDTH{1'b1}})
      next\_state = S\_FILL;
    else
      next state = S BACK;
```

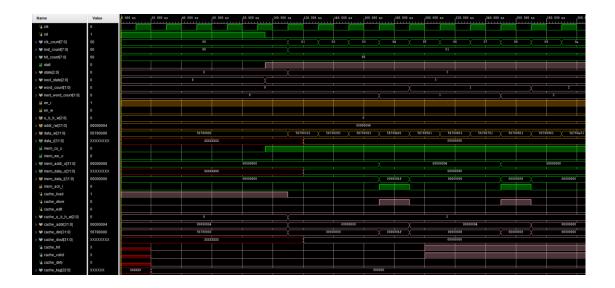
```
if (mem_ack_i)
      next_word_count = word_count + 1;
    else
      next_word_count = word_count;
  end
  S_FILL: begin
    if (mem_ack_i && word_count == {ELEMENT_WORDS_WIDTH{1'b1}})
      next_state = S_WAIT;
    else
      next_state = S_FILL;
    if (mem_ack_i)
      next_word_count = word_count + 1;
    else
      next_word_count = word_count;
  end
  S_WAIT: begin
    next_state = S_IDLE;
    next_word_count = 2'boo;
  end
 endcase
end
```

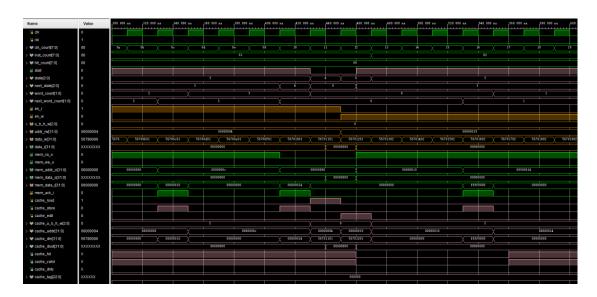
At last, we stall the pipeline when the cache is busy interacting with the memory.

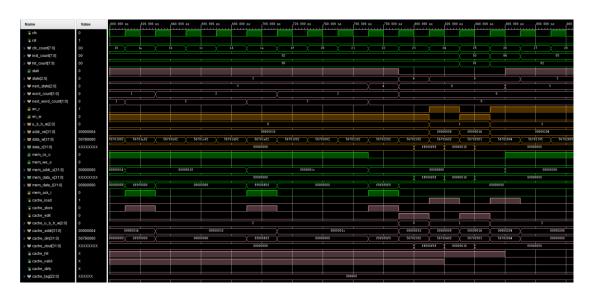
```
assign\ stall = \sim rst\ \&\ \sim (next\_state == S\_IDLE);
```

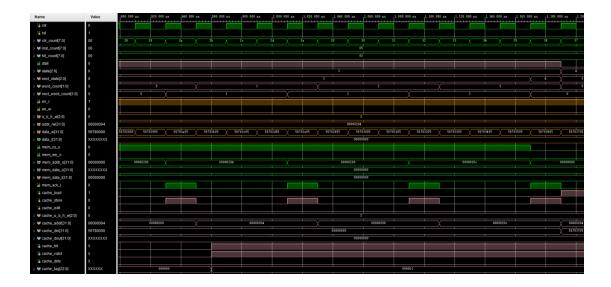
## 3.2Implementation results

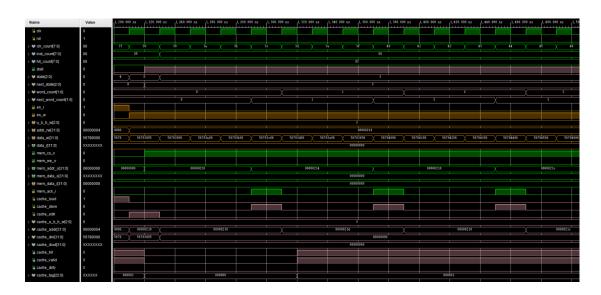
Here we record all our behavioral simulation results. Most will not be used for analysis.

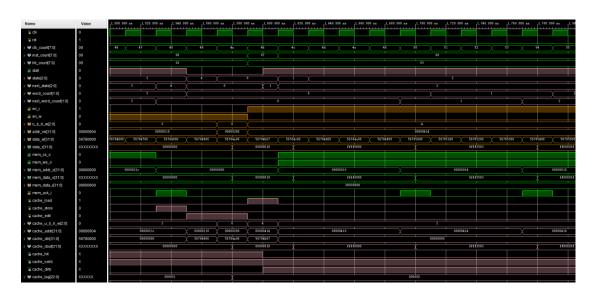


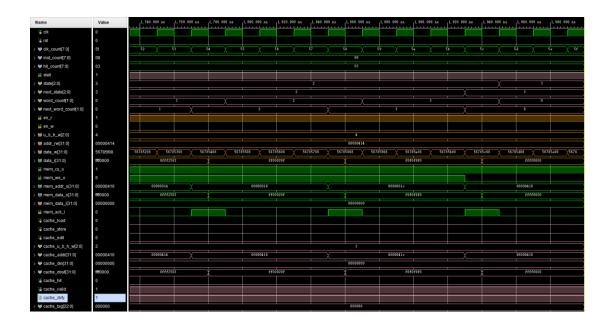








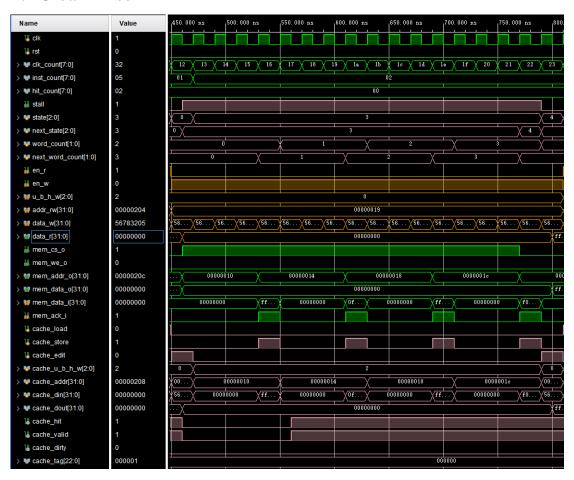




# 4 Analysis of the results

Here we only discuss some typical results.

### 4.1 Clean miss



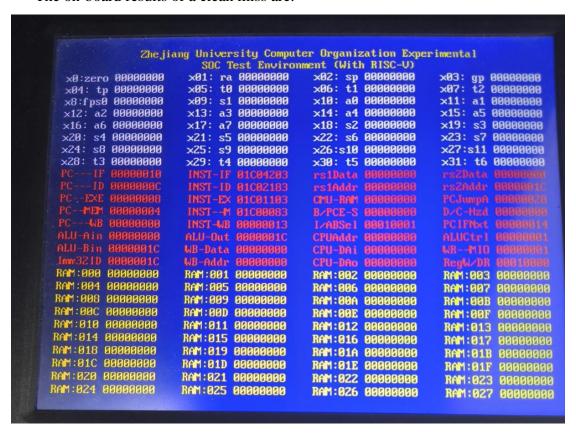
Consider the simulation in 450-810ns.

From 450ns, a **clean miss** happened. The state is transferred to **S\_FILL**, and the cache was gradually written (according to the **cache\_din** signal). At the end of **S\_FILL**, it transfers to **S\_WAIT**, and the cache output was given out.

Meanwhile, the whole pipeline is stalled.

After the process the data in memory is put in cache, we could see the process is successful.

The on-board results of a clean miss are:



S\_IDLE.

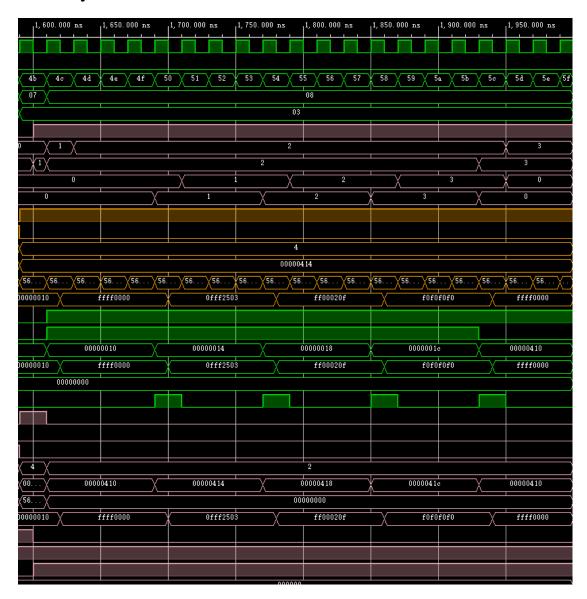
		ment (With RISC-U)	
×0:zero 00000000	×01: ra 00000000	x02: sp 00000000	×03: gp 00000000 ×07: t2 00000000
×04: tp 00000000	x05: t0 00000000	×06: t1 000000000 ×10: a0 00000000	x11: a1 00000000
x8:fps0 00000000 x12: a2 00000000	x09: s1 00000000 x13: a3 00000000	×14: a4 00000000	x15: a5 00000000
×16: a6 00000000	×17: a7 00000000	x18: s2 00000000	×19: s3 00000000
×29: s4 00000000	×21: s5 00000000	×22: s6 00000000	x23: s7 00000000
×24: s8 00000000	x25: s9 00000000	x26:s10 00000000	x27:s11 00000000
x28: t3 00000000	x29: t4 00000000	×30: t5 00000000	×31: t6 00000000
PCIF 00000010	INST-IF 01C04203	rs1Data 00000000	rsZData 000000000
PCID 0000000C	INST-ID 01C02183	rs1Addr 000000000	rs2Addr 000000010
PCEXE 00000008	INST-EX 01C01103	CMU-RAM 00030001	PCJumpA 00000028
PC-MEM 00000004	INSTM 01C00083	B/PCE-S 000000000	D/C-Hzd 000000000
. PCWB 0000000	INST-WB 00000013	I/ABSel 00010001	PCIFNxt 0000001
ALU-Ain 00000000	ALU-Out 0000001C	CPUAddr 00000000	ALUCtr1 00000000
ALU-Bin 0000001C	WB-Data 00000000	CPU-DA1 00000000	WR-MIO 0000000
Imm32ID 0000001C	WB-Addr 00000000	CPU-DAO 00000000	RegW/DR 0001000
RAM:000 00000000	RAM:001 00000000	RAM:002 00000000	RAM:003 0000000
RAM:004 00000000	RAM:005 00000000	RAM:006 00000000	RAM:007 0000000
RAM:008 00000000	RAM:009 00000000	RAM:00A 00000000	RAM:00B 0000000
RAM:00C 00000000	RAM:00D 00000000	RAM:00E 00000000	RAM:00F 0000000
RAM:010 00000000	RAM:011 00000000	RAM:012 00000000	RAM:013 0000000
RAM:014 00000000	RAM: 015 00000000	RAM:016 000000000	RAM:017 0000000
RAM:018 00000000	RAM:019 00000000	RAM:01A 00000000	RAM:01B 0000000
RAM:01C 00000000	RAM:01D 00000000	RAM:01E 00000000	RAM:01F 000000
RAM:020 00000000	RAM:021 00000000	RAM:022 00000000	RAM:023 000000
RAM:024 00000000	RAM:025 00000000	RAM:026 00000000	RAM:027 000000

The start of **S\_FILL** state. After this moment there are RAM automaton running to get the required data. The process will repeat 4 times so we omitted some graphs.

	Zhejia	ng Univers	sity Comput est Environ	er Organiz	ation Exper	imental	
	20000000	SUC TO	00000000000000000000000000000000000000	x02: sp	00000000	×03: gp 0	000000
x0:zero	00000000	x05: t0	00000000	×06: t1	00000000	x07: t2 0	
XV4: tp	00000000		00000000		00000000	×11: a1 0	
×12: a2	00000000	×13: a3	00000000	×14: a4	00000000	×15: a5 8	
×16: a6	00000000	×17: a7	00000000	×18: s2	00000000	×19: s3 (	
×29: s4	00000000	x21: s5	00000000	x22: s6	00000000	x23: s7 (	
	00000000		00000000	x26:s10	00000000	x27:s11 ( x31: t6 (	
	00000000		00000000		00000000	rsZData	
	00000010		01C04Z03		00000000	rsZbata	
	0000000C		01002183		00000000	PCJumpA	
	80000008		01C01103		00040000	D/C-Hzd.	
PCMEM			01000083		00000100	PCIFNet	
PCWB			00000013		00010001	ALUCTRI	
ALU-Ain			0000001C		00000000	WRMIO	
ALU-Bin			00000000		FFFFFFF	RegW/DR	
	0000001C		00000000		00000000	RAM:003	
RAM: 000		The second secon			00000000	RAM:007	
RAM: 004			00000000		00000000	RAM:00B	
	00000000	RAM:009	00000000		00000000	RAM:00F	
RAM:00C		RAM:00D	00000000		00000000	RAM: 013	
RAM:010			00000000		00000000	RAM:017	
	30000000	RAM: 015	00000000		00000000	RAM:018	
RAM:018			00000000		00000000		
	30000000		00000000		00000000	RAM:01F	
RAM:020 8		RAM:021	00000000		00000000	RAM:023	
RAM: 024 8	10000000	RAM: 025	00000000	RAM: 026	00000000	RAM: 027	nnnn

The **S\_WAIT** state at end.

## 4.2Dirty miss



Consider the result in 1590ns-2000ns.

In this period, a **dirty miss** is detected, as we found the **cache\_dirty** signal is valid. The state is first transferred to **S\_PRE\_BACK**, then **S\_BACK** for some clock cycles.

In these clock cycles, the memory in cache was written to the main memory according to the write-back principle. After that the state transferred to **S\_FILL**, and from that time on the process is similar to a **clean miss**.

The on-board results of a clean miss are:

```
Zhejiang University Computer Organization Experimental
                         SOC Test Environment (With RISC-U)
                                              x02: sp FFFFF0F0
                                                                      ×03: gp F0F0F0F0
                       ×01: ra ABCDE71C
x0:zero 00000000
                                                                      x07: t2 00000000
                                              x06: t1 00000000
                       x05: t0 0000F0F0
x04: tp 000000F0
                                              ×10: a0 00000000
                                                                      x11: a1
                                                                                00000000
x8:fps0 00000000
                       x09: s1 00000000
                                                                      ×15: a5 00000000
                       ×13: a3 000000000
                                              ×14: a4 00000000
x12: a2 000000000
                                              x18: s2 000000000
                                                                      x19: s3 00000000
×16: a6 00000000
                       ×17: a7 00000000
                       x21: s5 000000000
                                              x22: s6 00000000
                                                                      x23: s7 00000000
×20: s4 000000000
                                                                      x27:s11 00000000
                                              x26:s10 000000000
                       x25: s9 00000000
x24: s8 00000000
                                              x30: t5 00000000
                                                                      x31: t6 000000000
                       x29: t4 00000000
x28: t3 00000000
rs2Data 000000000
                        INST-IF FFDFF06F
                        INST-ID ØEDØ6813
                        INST-EX 41002403
                                              CMU-RAM 00000000
                                                                      PCJumpA 00000129
                                                                      D/C-Hzd 00000000
                                              I/ABSel 00010001
CPUAddr 00000000
                                                                      PCIFNxt 00000044
                       INST-WB 20002303
ALU-Out 00000400
                                                                      ALUCtrl 00000004
                       WB-Data 00000000
                                              CPU-DAi 00000000
                                                                      WR--MIO 00000001
                       WB-Addr 00000006
                                                                      RegW/DR 00010001
                                                                      RAM:003 00000000
                                               RAM:002 00000000
RAM:000 00000000
                       RAM:001 00000000
RAM:004
RAM:008
         00000000
000000000
                       RAM:005 00000000
                                               RAM:006 00000000
                                                                      RAM:007 0000F0F0
                                               RAM:00A 00000000
                                                                      RAM:00B 00000000
                       RAM:009 00000000
RAM:00C
         00000000
                                                                                00000000
                       RAM:00D 00000000
                                               RAM:00E 00000000
                                                                      RAM:00F
RAM:010 00000000
                       RAM:011 00000000
                                               RAM:012 00000000
                                                                      RAM:013 00000000
Ram:814 88888888
Ram:818 88888888
Ram:81C 8888888
Ram:820 8888888
Ram:824 8888888
                                              RAM:016 00000000
RAM:01A 00000000
RAM:01E 00000000
                       RAM:015 00000000
RAM:019 00000000
                                                                      RAM:017 00000000
                                                                      RAM:01B 00000000
RAM:01F 00000000
RAM:023 00000000
                           :01D 00000000
                            021 000000000
                                               RAM:022 00000000
                                                                      RAM:027 00000000
                            025 000000000
                                               RAM:026 00000000
```

#### S IDLE.

```
Zhe jiang University Computer Organization Experimental
                             SOC Test Environment (With RISC-U)
                                                                              x03: gp F0F0F0F0
                          ×01: ra ABCDE71C
                                                    ×02: sp FFFFF0F0
x0:zero 00000000
                                                    x06: t1 00000000
                                                                              x07: t2 00000000
x04: tp 000000F0
                          x05: t0 0000F0F0
                                                                              ×11: a1 00000000
                                                    ×10: a0 000000000
                          x09: s1 00000000
x8:fps0 000000000
                                                                              ×15: a5 00000000
                                                    ×14: a4 00000000
×12: a2 00000000
                          ×13: a3 00000000
                                                                              x19: s3 00000000
                                                    x18: s2 000000000
                          x17: a7 00000000
×16: a6 00000000
                                                    x22: s6 000000000
                                                                              x23: s7 00000000
x20: s4 00000000
                          x21: s5 00000000
                                                                              x27:s11 00000000
                                                    x26:s10 000000000
x24: s8 00000000
                          x25: s9 00000000
                                                                              x31: t6 00000000
                                                    x30: t5 000000000
x28: t3 00000000
                          x29: t4 00000000
PC--IF 98090949
PC--IF 98090949
PC--ID 9809093C
PC--EXE 98090934
PC--WE 98090939
ALU-Ain 98090999
ALU-Bin 98090999
PAM-200 98090999
                                                    rs1Data 000000000
                                    FFDFF06F
                                                                              rs2Addr 00000000
PCJumpA 00000123
D/C-Hzd 00000000
                          INST-ID ØEDØ6813
                                                    CMU-RAM 00010000
B/PCE-S 00000000
                          INST-M 40002383
INST-WB 20002303
                                                                              PCIFNxt 00000044
                          ALU-Out 00000400
                                                    CPUAddr 00000000
                                                                              WR-4MIO 90999991
RegW/DR 90919991
                          WB-Data 00000000
                                                    CPU-DAi 00008000
                          WB-Addr 00000006
                                                    CPU-DAO 00000000
                                                    RAM:002 00000000
                                                                              RAM:003 00000000
RAM:000 00008000
RAM:004 00000000
RAM:008 00000000
RAM:00C 00000000
                          RAM:001 00000000
                                                    RAM:006 00000000
                                                                              RAM:007 0000F0F0
                          RAM:005 00000000
                                                                              RAM:00B 00000000
                                                    RAM:00A 00000000
                          RAM:009 00000000
                                                    RAM:00E 00000000
                                                                              RAM:00F 00000000
                          RAM:00D 00000000
RAM:010 00000000
RAM:014 00000000
RAM:014 00000000
RAM:01C 000000000
RAM:01C 000000000
RAM:020 000000000
                                                                              RAM:013 00000000
RAM:017 00000000
                          RAM:011 00000000
                                                    RAM:012 00000000
                                                    RAM:016 00000000
                          RAM:015
                                    00000000
                                                    RAM:01A 00000000
                                                                               RAM:01B 00000000
                                19
                                    00000000
                                                    RAM:01E 00000000
                                                                               RAM:01F 00000000
                                    00000000
                                                    RAM:022 00000000
                                                                               RAM:023 00000000
                                                    RAM:026 00000000
                                    00000000
                                                                               RAM:027 00000000
```

```
Zhejiang University Computer Organization Experimental
                          SOC Test Environment (With RISC-V)
                                               x02: sp FFFFF0F0
                        ×01: ra ABCDE71C
                                                                       ×03: gp F0F0F0F0
 x0:zero 00000000
                                                         00000000
                                                                       ×07:
                                                                                 00000000
                        x05: t0
                                 0000F0F0
                                               ×06: t1
         000000F0
 x04: tp
                                                                       ×11: a1
                                                                                 00000000
                                               ×10: a0 000000000
         00000000
                        x09: s1
                                 00000000
x8:fps0
                                                                       ×15: a5
                                                                                РИВРИВИЯ
         00000000
                        ×13: a3
                                 00000000
                                               ×14: a4
                                                         00000000
 x12: a2
                                                                       x19: s3
                                                                                 00000000
         00000000
                        x17: a7
                                 00000000
                                               x18: s2
                                                         00000000
x16: a6
                                                                                 00000000
                                               x22: s6 00000000
                                                                       x23: s7
         00000000
                        x21: s5
                                 00000000
x20: s4
                                               x26:s10 00000000
                                                                       x27:s11 000000000
x24: s8 00000000
                        x25: s9
                                 00000000
                                               x30: t5 00000000
                                                                       x31: t6 00000000
                        x29: t4
                                 00000000
x28:
      t3 00000000
                                 FFDFF06F
                        INST-ID 0ED06813
INST-EX 41002403
                                               rs1Addr 00000000
CMU-RAM 00020004
B/PCE-S 00000000
                                                                       PCJumpA 00000123
                                 40002383
                                               I/ABSel 00010001
CPUAddr 00000000
                                                                        PCIFNxt
                        INST-WB 20002303
ALU-Out 00000400
                                                                       ALUCTRI
                        WB-Data 00000000
                                               CPU-Dai 000000000
ALU-Bin
                                               CPU-DAo
                        WB-Addr 00000006
                                               RAM:002 00000000
                                                                       RAM:003 00000000
RAM:000
RAM:004
                                 00000000
                        RAM:001
         00000000
                                               RAM:006
                                                                        RAM:007
                        RAM: 005
                                 00000000
                                                         00000000
                                                                                 0000F0F0
         00000000
RAM: 008
                        RAM:009
                                 00000000
                                               RAM:00A 00000000
                                                                        RAM:00B
                                                                                 00000000
                        RAM:00D
RAM:011
                                                                                 00000000
000000000
                                               RAM:00E
                                                                        RAM:00F
RAM:00C
RAM:010
         99999999
999999999
999999999
                                 00000000
                                                         00000000
                                               RAM:012 000000000
                                                                        RAM:013
                                 00000000
                        RAM: 015 00000000
                                               RAM:016 00000000
                                                                        RAM:017
                                                                                 00000000
RAM: 014
                                                                        RAM:01B 00000000
RAM:018 00000000
                             019 000000000
                                                RAM:01A 00000000
RAM:01C 00000000
RAM:020 00000000
RAM:024 00000000
                                 00000000
                                                RAM:01E
                                                         00000000
                                                                        RAM: 01F
                                 00000000
                                                         00000000
                                                                        RAM:023
                                                RAM: 022
                                 00000000
                                                RAM:026 00000000
                                                                        RAM:027
                                                                                 00000000
```

The start of **S\_BACK** state. RAM automaton runs to get the required data. The process will repeat 3 times so we omitted some graphs.

```
Zhejiang University Computer Organization Experimental
                         SOC Test Environment (With RISC-V)
                                              x02: sp
                                                       FFFFF0F0
                                                                     ×03: gp F0F0F0F0
                       ×01: ra ABCDE71C
 x0:zero 00000000
                                              x06: t1
                                                       00000000
                                                                     x07: t2
                                                                               00000000
 ×04: tp 000000F0
                                0000F0F0
                       x05: t0
                                                                     ×11: a1
                                00000000
                                              ×10: a0 00000000
                                                                               00000000
         00000000
                       x09: s1
 x8:fps0
                       ×13: a3 00000000
                                              ×14: a4
                                                       00000000
                                                                     ×15: a5
                                                                              00000000
         00000000
 x12: a2
                                                                     ×19: s3 00000000
                                                       00000000
                                              x18: s2
 ×16: a6
         00000000
                       ×17: a7
                                00000000
                                                                     x23: s7 00000000
                                              x22: s6
                                                       00000000
                       x21: s5
                                00000000
         ВВВВВВВВВ
 x20: s4
                                              x26:s10 000000000
                                                                     x27:s11 00000000
x24: s8
         00000000
                       x25: s9 00000000
                                                                     x31: t6 00000000
                                              ×30: t5
                                                       00000000
x28: t3 00000000
                       x29: t4 00000000
                                              rs1Data 00000000
                                FFDFF06F
                                              rs1Addr 00000000
CMU-RAM 00030001
B/PCE-S 00000000
                       INST-EX 41002403
INST--M 40002383
INST-WB 20002303
                                              CPUAddr 00000000
CPU-Dai 00008000
                                                                      ALUCtr1
                       ALU-Out 00000400
                                00000000
                                              CPU-DAo
                                                       00000000
                                                                      RegW/DR
                                                                      RAM:003 00000000
                                              RAM:002 00000000
RAM: 000
         00000000
                       RAM:001 00000000
                                              RAM: 006
                                                                      RAM:007
RAM:004
RAM:008
         00000000
                       RAM:005
                                00000000
                                                       00000000
                                                                      RAM:00B
                                00000000
                                              RAM: 00A
                                                       00000000
                                                                               00000000
                       RAM:009
RAM: 00C
         00000000
                                              RAM:00E
                                                       00000000
                                                                      RAM: 00F
                                                                               00000000
                       RAM: 00D
                                00000000
RAM:010 00000000
                       RAM:011
                                00000000
                                              RAM:012
                                                       00000000
                                                                      RAM:013
                                                                               00000000
RAM:014 00000000
RAM:018 00000000
RAM:01C 00000000
                       RAM: 015
                                              RAM:016
                                                       00000000
                                                                      RAM:017
                                00000000
                                              RAM:01A 00000000
                                                                      RAM: 01B
                                                                               00000000
                                00000000
                                              RAM:01E 00000000
                                                                      RAM:01F 00000000
RAM:020 0000000
                                              RAM:022 00000000
                                00000000
                                                                      RAM:023 00000000
                                 00000000
                                              RAM:026 00000000
                                                                      RAM:027 00000000
```

The start of **S\_FILL**, similar to above.

	Zhe jiang l	Inivers	ity Comput	er Organiz ment (With	ation Expe	rimental	
x0:zero 00 x04: tp 00 x8:fps0 00 x12: a2 00	00000F0 ×6 0000000 ×6 0000000 ×1	11: ra 15: t0 19: s1 13: a3	ABCDE71C 0000F0F0 00000000 00000000	x02: sp x06: t1 x10: a0 x14: a4	FFFFF0F0 00000000 00000000 00000000 000000	x03: gp   x07: t2   x11: a1   x15: a5 x19: s3	000000000 000000000
x16: a6 06 x20: s4 06 x24: s8 06	0000000 ×	21: s5 25: s9	00000000 00000000 00000000	x22: s6 x26:s10	00000000 00000000	x23: s7 x27:s11 x31: t6	00000000 000000000 000000000
x28: t3 00 PCIF 00 PCID 00 PCEXE 00	9000040 II 900003C II	NST-IF	FFDFF06F 0ED06813 41002403	rs1Data rs1Addr CMU-RAM	00000000 00000000 00040000	rsZAddr PCJumpA	000000000 00000000 00000123
PCMEM 00 PCWB 00 ALU-Ain 00	0000034 II 0000030 II 0000000 A	NSTM NST-WB LU-Out	40002383 20002383 00000400	I/ABSel CPUAddr	00000100 00010001 00000000	PCIFNxt ALUCtrl	000000000 000000004 000000001
	00000ED W	B-Addr AM:001	00000000 00000000 00000000	CPU-DAo RAM: 002	00000000 00000000 00000000	RegW/DR RAM:003	00000000 0000F0F0
and the second second second	0000000 R 0000000 R		00000000	RAM:00A RAM:00E	00000000 00000000	RAM:00B RAM:00F RAM:013	999999999 999999999
RAM:014 00 RAM:018 00	3000000 R		80808080 80808888	RAM:016 RAM:01A RAM:01E	00000000 000000000 000000000	RAM:01F	00000000
RAM:020 01 RAM:024 01			00000000		00000000		00000000

S\_BACK.

## 5 Discussion and Conclusion

### 5.1 Problems

## 5.1.1 Problems concerning the Experiment Guides

#### Problem 1. Program Difference

The program used in the experiment guilds are not the similar as given in the project and waveform simulation. That original program can differ **lw**, **lb**, ... operations, but the given program cannot.

We have modified the given program to overcome this difference, so some of our data may not be similar to other groups.

#### 5.2 Achievements and conclusion

In this experiment, I implemented a cache controller based on the cache we wrote in lab-3, and learned the knowledge about CPU-Memory communication. Overall the experiment is successful.