洲江水学

本科实验报告

计算机体系结构
姜雨童
计算机科学与技术学院
计算机科学与技术
3220103450@zju.edu.cn
1369218489
18867766468
王小航
2024年 11月 17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	
实验项目名称	: Lab04 Pipelined	I CPU with Cache		
学生姓名:	<u>姜雨童</u> 学号: _	33220103450]组学生姓名:_	
实验地点: _	玉泉曹西 301	实验日期: <u>_202</u> 4	年 <u>11</u> 月	17_日

一、目标与原理、

1-1 实验目的

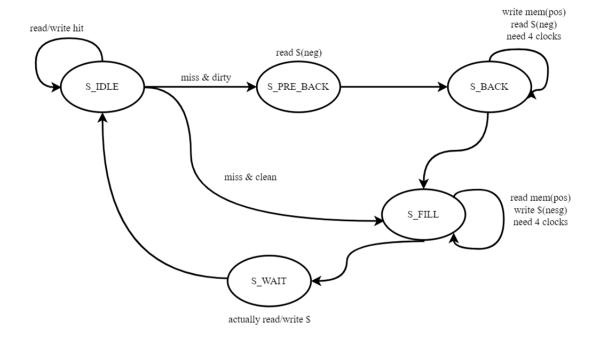
- Understand the principle of Cache Management Unit (CMU) and State Machine of CMU.
- Master the design methods of CMU and Integrate it to the CPU.
- Master verification methods of CMU and compare the performance of CPU when it has cache or not.

1-2 实验原理

本实验延续了实验三(实现 cache),需要实现 cache controller,即 cache 的管理单元。它的实现就是一个拥有五个状态的有限状态机(见下图):

- 1. S_IDLE:初始状态,如果 hit,则继续维持在此状态;
- 2. S_PRE_BSCK:和 S_BACK 一起构成 miss 时需要写回内存(即 dirty)的状态转移;
- 3. S_BACK:miss 时需要写回内存时的状态转移,之后去到状态 S_FILL;
- 4. S FILL:miss 且不需写回内存/已经写回内存后的状态,从内存中读数据到 cache;
- 5. S_WAIT:miss 处理后重新执行指令。

在实现 cache controller 的状态机后,还需要将其接入 pipelined CPU。因此当 cache 不能较快速地返回数据时,需要对 pipeline 进行 stall 操作,因此在 cache controller 中还需要对 stall 进行判断。



二、操作方法与实验步骤

2-1 状态机

本实验的整体框架和代码均已给出,只需要填几个空。而根据上述对于状态机的分析及状态转移图,可以轻松写出状态转移的这几个空,故这里不做过多分析。另外还有几个空是针对计数器的,由于项目中不论是从 memory 读入 cache 还是从 cache 写回 memory 都是每次一个 word,因此当收到 ack 信号(成功读入/写回时),计数器加一,否则维持当前值。附源代码:

```
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'b00;
   end
   S_PRE_BACK: begin
       next_state = S_BACK;
       next word count = 2'b00;
   end
```

```
S_BACK: begin
                   if (mem_ack_i && word_count == {ELEMENT_WORDS_WIDTH{1'b1}})
2'b11 in default case
                       next_state = S_FILL;
                   else
                       next_state = S_BACK;
                   if (mem_ack_i)
                       next_word_count = word_count + 2'b01;
                   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 + 2'b01;
                   else
                       next_word_count = word_count;
               end
               S_WAIT: begin
                   next_state = S_IDLE;
                   next_word_count = 2'b00;
               end
           endcase
```

2-2 判断 stall

当需要状态转移到 S_{IDLE} 以外的状态时,说明需要写回 memory/读入 cache,因此要对流水线进行 stall 操作,代码如下:

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

三、实验结果与分析

3-1 仿真

Simulation

inst.v

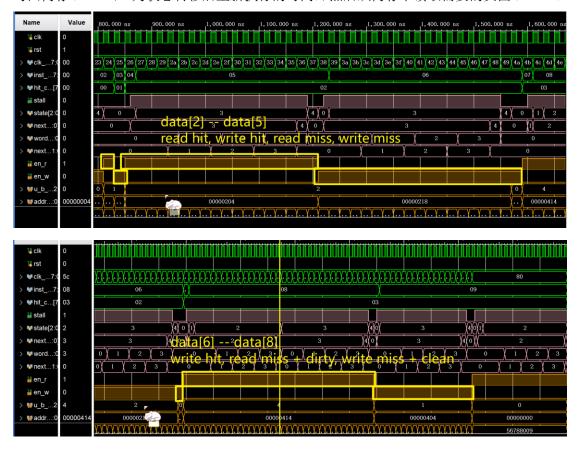
```
reg [39:0] data [0:9];
initial begin
    data[0] = 40'h0_2_00000004; // read miss
                                                           1+17
    data[1] = 40'h0_3_00000019; // write miss
                                                           1+17
    data[2] = 40'h1_2_00000008; // read hit
                                                           1
    data[3] = 40'h1 3 00000014; // write hit
                                                           1
    data[4] = 40'h2_2_00000204; // read miss
                                                           1+17
    data[5] = 40'h2_3_00000218; // write miss
                                                           1+17
    data[6] = 40'h0_3_00000208; // write hit
                                                           1
    data[7] = 40'h4_2_00000414; // read miss + dirty
                                                           1+17+17
    data[8] = 40'h1_3_00000404; // write miss + clean
                                                           1+17
    data[9] = 40'h0;
                              // end
                                                          total: 128
end
assign
    u_b_h = data[index][38:36],
    valid = data[index][33],
   write = data[index][32],
    addr = data[index][31:0];
```

指令 0 和 1 分别是 read miss 和 write miss,如下两图所示,由于 miss 每条指令都需要 1+17 (4*4+1,需要读/写 4 次 word,最后加一是状态转移后需要重新执行指令)个时钟周期的时间,中间有四次 ack 信号表示读取/写入 word 成功。(图 2 橙色框为 read miss,蓝色框为 write miss。)



其他指令的分析不在此赘述, 仿真波形截图如下:

其中 read miss dirty 需要 1+17+17 的时间是因为 read miss 的 cache 为 dirty 时首先需要将其写回内存(4*4+1,1 为状态转移后重新执行的时间),然后从内存中读取需要的页面(4*4+1).



3-2 上板

一共发生六次 miss, 根据 32 位地址的划分,可以算出这些指令对应地址的 tag, index 和 offset:

No	Tag	Index	Offset
1	0	1	15
6	1	1	0
9	0	0	0
12	1	0	0
13	2	0	0
14	2	1	0

分析代码的实现可知, index 为奇数对应 set1, 偶数对应 set0; tag 奇数对应 line1, 偶数对应 line0, 因此可以得到指令右侧标注的 set 和 line。

附指令如下:

R O M

NO.	Instruction	Addr.	Label	ASM	Comment
0	00000013	0	start:	addi x0, x0, 0	
1	01c00083	4		lb x1, 0x01C(x0)	# F0F0F0F0 in 0x1C # FFFFFFF0 miss, read 0x010~0x01C to set 1 line 0
2	01c01103	8		lh x2, 0x01C(x0)	# FFFFF0F0 hit
3	01c02183	С		lw x3, 0x01C(x0)	# F0F0F0F0 hit
4	01c04203	10		lbu x4, 0x01C(x0)	# 000000F0 hit
5	01c05283	14		lhu x5, 0x01C(x0)	# 0000F0F0 hit
6	21002003	18		lw x0, 0x210(x0)	<pre># miss, read 0x210~0x21C to cache set 1 line 1</pre>
7	abcde0b7	1C		lui x1 0xABCDE	
8	71c08093	20		addi x1, x1, 0x71C	# x1 = 0xABCDE71C
9	00100023	24		sb x1, 0x0(x0)	<pre># miss, read 0x000~0x00C to cache set 0 line 0</pre>
10	00101223	28		sh x1, 0x4(x0)	# hit
11	00102423	2C		sw x1, 0x8(x0)	# hit

R O M

NO.	Instruction	Addr.	Label	ASM	Comment
12	20002303	30		lw x6, 0x200(x0)	<pre># miss, read 0x200~0x20C to cache set 0 line 1</pre>
13	40002383	34		lw x7, 0x400(x0)	<pre># miss, write 0x000~0x00C back to ram, then read 0x400~40C to cache set 0 line 0</pre>
14	41002403	38		lw x8, 0x410(x0)	<pre># miss, no write back because of clean, read 0x410~41C to chache set 1 line 0</pre>
15	0ed06813	3c	loop:	ori x16, x0, 0xED	# end
16	ffdff06f	40		jal x0, loop	

首先指令 1 处,访问未访问过的 cache,是 miss read,需要从内存中读取:

指令1进	те прининине ге прининине	INST 1F 01004203	rsiData 98009900 rsiData 98009900	rs20ata 000000000 rs20ddr 00000010
入 mem	ГС Г⊃Е ИВИВИВИВ	INST EX 01001183	CMU-RAM BESSESSES	PGJumpa 00000028 DZC-Hzd 00000000
阶段,开	рт РТР нининяна	INST W 8108883	BARCE S MUNICIPAL I ARSET MONTHUM	PCTFMxt 88888814
始读	алл түй ининини	ALU Out 8888881C	CPUAddr 88888888	ALUCES BURGARDES
	1.0 Rin 8888881C	WB Data 98989888 WB Oddr 98988888	CPU-DAI 00000000 CPU-DAO 00000000	RegW/DR 00010000
1/	IDE-00 00000013	CODE-01 01C00083	CODE-02 01C01103	CODE-03 01C0Z183
(1+17)	IDE 04 01C04203	CODE-05 00000000	CODE-06 00000000	CODE-07 00000000
读第一个	РС 11 инивиния РС 10 инивиния	INST IF 81084283 INST ID 81082183	rs10ata 888888888 rs16dr 88888888	rsZData USUSUSUS rsZAddr 8888881C
word	PC EXE HUHHHHHH	INST EX BICBIES	CPU-Rem 88838881	PCJumpa @@@@@28
	PER BRIBHARIA	INST # 81088883	B PCE S BBBBBBBB	DAC-HAR BUBBBBBB PCIFNAL BUBBBBB14
计时器第	"(• В нининии в на вининии	INST WB 00000013	I/ABSel 88818881 CPUnddr 88888888	ALUCTAL BESSESSE
一拍	LII Ria BEBBBBIC	t.B. Data 88888888	CPU-Dai 00000000	WR FTO BESSESSE
	or ELD BRBBBBBC	WB-nddr 88888888	CPU-Dao 80888888	RegW/DR 88819888
1/17	IDE-00 00000013	CODE-01 01C00083	CODE-02 01C01103	CODE-03 01C0Z183
	DE -04 01C04203	CODE-05 00000000	CODE-06 00000000	CODE-07 00000000 CODE-0B 00000000
	DE АВ ИНИИИИ	CODE-89 88888888	CODE-0A 00000000	rszData Generale
计时器第	PC II BEBEBEE	INST-IF 01C04203	rsiData 00000000 rsiAddr 00000000	rsznadr namanate
二拍	Pr 12:1: ниивинив	INST EX BICBIES	CMU-RAM BBB BBBB2	Редитра вовового
	PER PER BEREBURA	INST # BICBBBB3	BAPCE S MMMMMMM	D-C-H24 BUBBBBBB
1/17	ес кл нанинини	INST WB 80888813	1/ABSel 00010091 CPUnddr 000000000	PCIFNAL BBBBBBB14
	LU Din 88888888	OLU Out 9888981C	CPU-DAI BBBBBBBB	WR - Fritti BBBBBBBB
	or 321D BBBBBBB1C	WB Addr 88888888	CPU-DAG 00000000	RegN/DR 88818888
	IDE-00 00000013	CODE-01 01C00083	CODE-02 01C01103	CODE-03 01C02183
	DE 04 01C04Z03	CODE-05 00000000	CODE-06 00000000	CODE-07 00000000
	er oo ooooooo	cont ag aggagggg	CONF_0A QQQQQQQQQ	CODE-AR ААААААА

计时器第 三拍 1/17	17 11 HIBBIRD 18 19 10 HEBBIRD 18 19 12 HEBBIRD 19 10 12 HEBBIRD 18 11 ATT HIBBIRD 18 11 ATT HIBBIRD 10 10 E HE HIBBIRD 10 10 E HE HIBBIRD 13 10 E HE HIBBIRD 3 10 E HIB	INST IF BICHSZHS INST ID BICHSZHS INST EX BICHSHSS INST EX BICHSHSS INST WE GROUNDS INST WE GR	PSIDALA GREGORIOS PSIANT GREGORIOS PSIANT GREGORIOS BYPCE-S GROGORIOS IVARSE 1 GREGORIOS IVARSE 1 GREGORIOS CPU-DAG GREGORIOS CPU-DAG GREGORIOS CODE-92 91 C91193 CODE-94 99999999 CODE-95 99999999	PSZDATA MUGDOMAB PSZDATA MUGDOMAB PCIDDOM BRIGGOZD DZC M24 GRBBBBBB PCIFRST BRIGGOZD MLUCERI BRIGGODI UR PUD BRIGGODI CODE-03 BIC92183 CODE-07 BRIGGODI CODE-08 GRBBBBBB CODE-08 GRBBBBBB CODE-08 GRBBBBBB CODE-08 GRBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
计时器第四拍 (后续重 复3遍)	DE-00 00000013 DE-04 01C04203	INST IF #1084281 INST ID #1082183 INST ID #1081183 INST # #1081883 INST WE #8888813 ALU INAL #88888810 WE hada #8888888 WE hada #8888888 CODE-81 #108883 CODE-81 #108883 CODE-85 #8888888	rs10ata rs1atar sususissis Gru-Rri B/PCE S sususissis 1/ABSe/J sususissis CPU-Data sususissis CPU-Data sususissis CPU-Data sususissis CPU-Data sususissis CPU-Data sususissis CPU-Data sususissis CODE-82 81C81183 CODE-84 88888888	N.2DATA BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
状态转移 重新执行 指令 1/17	>28: 13 00000000 РС 17 ИНВИВИНИЕ РС 15-1 ИНВИВИНИЕ РС 15-1 ИНВИВИНИЕ РС 4-1 ИНВИВИНИЕ ГС 4-1 ИНВИВИВИИ ПЛИ В 11 ИНВИВИИ ГС ГОТЕ 00 ИНВИВИВИЗ СОДЕ 04 01004203 СОДЕ 08 0000000		x38: t5 888888888 realbata 888888888 realbata 888888888 realbata 888888888 Reports 8881 8881 character 88888888 realbata 888888 code-86 88888888 code-86 88888888 code-86 88888888 realbata 88888888 code-86 888888888	x31: t6 88888888888888888888888888888888888
进入下一 条指令	Pr	INST IF 01605203 INST ID 01605203 INST ID 01602103 INST IN 01602103 INST IN 016000003 INST IN 016000003 INST IN 0160000003 INST IN 0160000003 INST IN 01600000000000000000000000000000000000	PSIDATA CHRISTISH PSITATA CHRISTISH PSITATA SHERRISHS B. POE S. SERRISHS I. CPUNIAI SHERRISHS I. CPUNIAI SHERRISH SHE	#520ata 88888888 #520ata 8888881C #531aam 8888882C 8757824 88888888 #6117824 88888888 #6117824 88888888 #6117824 888888883 #6117824 888888883

指令 6 访问同一 set 不同 line,指令 9 访问不同 set,指令 12 访问也与前面都不相同,因此也都是 miss read,需要从内存中读取数据到 cache (过程类似,不放上板截图做赘述)。

指令 13 访问 set0, line0,与指令 9 是同一个块,但是二者 tag 不同,因此 miss。且指令 9 中将内容写入 cache,因此该块是 dirty 的,需要将数据写回内容:

入 mem 阶段,开始	######################################	X20:S18 BBBBBBBB X27:S X30: t5 BBBBBBBBB X31: rs2b; rs1add BBBBBBBBB BBBBBBBBBB BBBBBBBBBB
-------------	--	--

to the first tests	и вининия	- 5555600	x30: t5 00000000	~L1 .;
写回第一	те принцина	INST IF FFDFF86F	rs10ata 66666666	×31:
个 word	те 1≥×1. Вивини зн	INST 10 0ED06813	PSIAddr Bassassas	rszh
	ГС № ВИНИВИЗА	INST # 48882383	CPU ROM MAGARMAN	rszn
计时器第	BE BRIBBB 34	INST &B 20002303	B PCE-S BIBBIBIBIBI	PCJu
一拍	III nin иниинив	ALU-Out 88888488	L'ABSEL BEBLEBBEL	PCIP
	II Rin инини418	WB Data 88888888	Crunddr agaggggg	MLUC
1/17	изгла ининиваер	WB Addr 88888886	CPU-DA i BABBBBBBB	WR
	DE-80 00000013	CODE-01 01C00083	CPU-DAO ПЯПЯПИНА	Regio
	E-04 01C04203	CODE-05 01C05283	CUDE-02 01C01102	CODE
	E-08 71C08093	CODE-09 00100023	CONF-RE 51885885	CODE
	:-0C 20002303	CODE-8D 48882383	CUDE-BA ARTA1222	CODE
写回第一	SECTION SECTIO	INST IF PERFERE	ADD. TO MUMARADA	×31:
	10 инивинас	INST ID BEDBERG		rsZDe
个 word	ЮТ: ининия зв МЭТ: ининия за	1007 EX 41882480	PS Triddy Balaisiaisia	rsZna
计时器第	ыл ининиизи	INST # 48882202	CMU RAM GGGZGGG4	PCJu
	Divis Differences	THE T WILL CHIEF THE	BYPCE-S SUSSESSES	B/C-I
二拍	III A STATE OF THE PROPERTY OF THE PARTY OF	HLU Uut BBBBBBBBB	I/ABSel 00010001	PCIP
	All proposes	wb Data BBBBBBBBB	CPUAddr 00000000	MLUC
	-99 000000	WD Hadr MANAGARA	CPU-DAI 8888888C	WR-4
	RA RICOADO	JULE-01 01C00000	CPU-DAO 00000000	Regul
	99 71000000	UDE-05 01C05283	CODE-NZ N1C01102	CODE
	1C 20000000	UNE-89 BRIBBASS	CONF-AP SINUSABO	CODE
	IC 20002303 CI	IDF-0D 4000000	CODE-0A 00101223	COPP
写回第一		מאאאאאאאא בי יביי		X47::
	т П инивина п пинивина	INST IF PEDERAGE	×30: t5 ппппппп	
个 word	РС 12 Е ВИВИНИ И В 12 ВИВИНИ И	INST IF PEDEFUGE	x30: t5 00000000	x27: x31: rs20
	т транивана и противората про	INST IF PEDEFROR INST ID BEDBEBLIJ INST IX 11882483	x30: t5 00000000 railata 60060000 railata 60060000 railata 60060000 railata 60060000	x27:: x31: rs20 rs28
个 word 计时器第	РС ПР инивинаци РС ПР инивинаци РС ПР инивинаци РС И инивинаци	INST IF PEDEFECT INST ID BEDREADS	x30: t5 88888888 rsibata 88888888 rsibata 88888888 CMU-Ram 88828885 B>PCE-S	x27: x31: rs20 rs2n PCdu
个 word	РС 17 инивиналь РС 15-Е инивиналь РС 15-Е инивиналь РС 15-Е инивиналь РС 16-Е инивиналь РС 17-Инивиналь	INST IF PROPERTY INST ID HERBERTS INST INST INST INST INST INST INST IN	x30: t5 BBBBBBBB rsiData BBBBBBBB rsindar BBBBBBBB CMU-Rom BBB2BBBB B/PCE-S BBBBBBBB L/ABSel BBBBBBBB	x27: x31: rs2b rs2c PCdu b/C-
个 word 计时器第	PC ID BEBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB	INST IF PEDEFROR INST ID REDRIGHTS INST IX 41882483 INST W 48882383 INST W 28882383 INST W BRRRRASS	x30: t5 BBBBBBBB rsilata BBBBBBB rsindtr BBBBBBBB CMU-Ram BBBBBBBB B/PCE-S BBBBBBBBB I/ABSel BBBBBBBB CPUnddr BBBBBBBB	x27: x31: rs2b rs2c PCdu BxC- PCIF
个 word 计时器第	PC ID BEBERRA PC IDE BEBERRA PC IDE BEBERRA PC LE BEBERRA PC LE BEBERRA BLU DIN BEBERRA LU DIN BEBERRA BUT BUT BEBERRA BUT BUT BEBERRA	INST IF PEDEFROR INST ID REDRIGHTS INST ID 41882483 INST IV 41882383 INST W 28882383 ALU Out 88888488 WB-Data 88888888	x30: t5 00000000 rsibata 00000000 rsindar 00000000 00U-Rom 000000000 BZPCE-S 00000000 IZABSEI 00010001 CPUAdar 00000000	x27: x31: rs2b rs2r PCdu b/c- PCTF ALUC
个 word 计时器第	TE HUBBHARA TO THE BRUBHARA TO TOP BRUBHARA TO TOP BRUBHARA TO THE BRUBHARA THE	INST IF PEDEFROF INST ID REDRESSTS INST EX 41882483 INST W 48882383 INST W 28882383 INST W 888888888	×30: t5 00000000 rsibata 60000000 rsindar 60000000 600-000 60000000 BZPCE-S 600000000 IZABSEI 60010001 CPU-DAI 60000000 CPU-DAI 600000000	×27: ×31: rs20 rs2n PCdu D/C- PC II: ALUC WB-
个 word 计时器第	ПР ИНИВИНАЦИ ПО 110 ИНИВИНИ 10 ПО 15/Е ВИВИВИИ 30 ПО 15/Е ВИВИВИИ 30 ПО 16/Е ВИВИВИИ 30 ПО 16/Е ВИВИВИИ 30 ПО 16/Е ВИВИВИИ 30 ПО 16/Е ВИВИВИИ 41 ПО 16/Е ВИВИВИ 41 ПО 16/Е ВИВИВИИ 41 ПО 16/Е ВИВИВИВИИ 41 ПО 16/Е ВИВИВИИ 41 ПО 16/Е ВИВИВИ 41 ПО 16/Е ВИВИВИ 41 П	INST IF PEDEFROF INST ID REDROBATA INST IX 11892483 INST IX 11892483 INST IX 18882383 INST IX 28882383 INST IX 98888488 INST IX 98888488 INST IX 98888488 INST IX 988888888 INST IX 9888888888 INST IX 9888888888 INST IX 9888888888 INST IX 98888888888 INST IX 988888888888888888888888888888888888	×30: t5 090000000 rsibata didebined designed de	x27: x31: rs2b rs2c PCJu b/c PCJE nLUC WR- Regt
个 word 计时器第	ПЕ ИНПИВИНА ПО ИНИИНИ ПО ПО ПО ИНИИНИ ПО	INST IF PEDEFROF INST ID REDROBATA INST ID REDROBATA INST ID 41882483 INST W 41882383 INST W 48882383 INST W 88888488 INST W 88888888 INST W 88888888 INST W 888888888 INST W 888888888 INST W 888888888 INST W 8888888888 INST W 8888888888 INST W 8888888888 INST W 88888888888 INST W 888888888888 INST W 88888888888888888888888888888888888	X30: t5 00000000 rsiData 00000000 rsiData 00000000 rsiData 00000000 selection 0MU-Rem 000020000 0MU-Rem 00002000000 0MU-Rem 0000200000 0MU-Rem 0000200000 0MU-Rem 0000200000000000000000000000000000000	x27: x31: rs2n rs2n PCJu D/C PC IF nLUC WB- Hegd CODE
个 word 计时器第	ПР ИНИВИНАЦИ ПО 110 ИНИВИНИ 10 ПО 15/Е ВИВИВИИ 30 ПО 15/Е ВИВИВИИ 30 ПО 16/Е ВИВИВИИ 30 ПО 16/Е ВИВИВИИ 30 ПО 16/Е ВИВИВИИ 30 ПО 16/Е ВИВИВИИ 41 ПО 16/Е ВИВИВИ 41 ПО 16/Е ВИВИВИИ 41 ПО 16/Е ВИВИВИВИИ 41 ПО 16/Е ВИВИВИИ 41 ПО 16/Е ВИВИВИ 41 ПО 16/Е ВИВИВИ 41 П	INST IF PEDEFROF INST ID REDROBATA INST ID REDROBATA INST ID 41882483 INST W 41882383 INST W 48882383 INST W 88888488 INST W 88888888 INST W 88888888 INST W 888888888 INST W 888888888 INST W 888888888 INST W 8888888888 INST W 8888888888 INST W 8888888888 INST W 88888888888 INST W 888888888888 INST W 88888888888888888888888888888888888	×30: t5 00000000 rsibata 60000000 rsindar 60000000 600-000 60000000 BZPCE-S 600000000 IZABSEI 60010001 CPU-DAI 60000000 CPU-DAI 600000000	x27: x31: rs2b rs2c PC de B/C- PC II: nLuc WR- Regt CODE
个 word 计时器第 三拍	ПЕ ИНПИВИНА ПО ИНИИНИ ПО ПО ПО ИНИИНИ ПО	INST IF PEDEFROF INST ID REDROBATA INST IN ALBREMANA INST IF PEDEFROF INST IN ALBREMANA INST IF PEDEFROF INST IN ALBREMANA INST IF PEDEFROF INST IN ALBREMANA IN	X30: t5 00000000 rs1Data Hiddendon rs1Data Hiddendon rs1Data Hiddendon rs1Data Hiddendon Richard Richa	x27: x31: rs2n rs2n PCJu D/C PC IF nLUC WB- Hegd CODE
个 word 计时器第	ПР ИНИВИНА НЕ ПР ИНИВИНА НЕ ПР ИНИВИНИ НЕ ПР ИНИВИ НЕ ПР ИНИВИНИ НЕ ПР ИНИВИТИТЕТ НЕ ПР ИНИВИТЕТ НЕ ПР ИН	INST IF PEDERAGE INST ID REDRESSION INST ID REDRESS	X30: t5 00000000 rsilata 60000000 rsilata 60000000 rsilata 60000000 rsilata 60000000 rsilata 60000000 rsilata 60000000 significati 60000000 rsilata 60000000 significati 60000000 significati 60000000 college 7000000000 college 70000000000 college 70000000000 college 700000000000000000000000000000000000	x27: x31: rs2n rs2n PCdu D/C PCTE nLUC WR- Regu CODE CODE
个 word 计时器第 三拍	ПР ИНИВИНА НЕ ПР ИНИВИНА НЕ ПР ИНИВИНИ НЕ ПР ИНИВИРИ НЕ ПР ИНИВИТИТЕТТИ НЕ ПР ИНИВИТЕТТИ НЕ ПР ИНИВИТИТЕТТИ НЕ ПР ИНИВИТИТЕТТИ НЕ ПР ИНИВИТЕТТИ НЕ	INST IF PEDERGOT INST ID REDROGRAM INST IN ALBREAGER INST IN ALBREAGER INST IN ALBREAGER INST INST IN BERNER INST IN BERNER INST IN BERNER INST IN BERNER INST ID REDROGRAM IN	X39: 15 09000000 rs10ata postulata postulata postulada osciologa comunication de la comunicación de la comun	x27: x31: rs2h rs2h PCdu D/C PCTE nLUC WR- Regt CODE CODE CODE
个 word 计时器第 三拍 写回第一 个 word	ПР ИВИВИВНА В ПР ПР ИВИВИВНА В ПР ПР ВВИВИВИЗИ В ПР ПР ВВИВИВИЗИ В ПР ПР ПР ВВИВИВИ В В ПР ПР ВВИВИВИ В В ПР ПР ВВИВИВИ В В ПР ВВИВИВИ В В ВВИВИВИ В В ВВИВИВИ В	INST IF PEDERGOF INST ID BEDBGB13 INST IX 11892493 INST IX 11892493 INST IX 11892493 INST IX 18982383 INST IX 189889888 INST IX 189889883 INST IX 18988983 INST IX 18988983	X30: t5 09000000 rstData rstDa	x27: x31: rs20 rs2n PCdu D/C PCTE nLUC WR- Regu CODE CODE CODE x31: rs20 rs2n
个 word 计时器第 三拍	ПЕ ИВИВИНА ПО ПЕВИВИНА ПО ПЕВИВИНИ ПО ПЕВИВИВИИ ПО ПЕВИВИВИИ ПО ПЕВИВИВИ ПО ПЕВИВИ П	INST IF PEDEFROF INST ID REDROBATA ALU-OUT REROBARA ALU-OUT REPORTAN INST ID REDROBATA	X30: t5 09000000 rstData rstDa	x27: x31: rs20 rs2n PCdu D/C- PC IE nLUC WR- Rege CODE CODE CODE x31: rs2n rs2n rs2n
个 word 计时器第 三拍 写回第一 个 word 计时器第	ПР ИНИВИНА НЕ ПР ИНИВИНА НЕ ПР ИНИВИНИ НЕ ПР ИНИВИ НЕ ПР ИНИВИНИ НЕ ПР ИНИВИНИЯ НЕ ПР ИНИВИВИЛИ НЕ ПР ИНИВИЛИ НЕ ПР ИНИВИВИЛИ НЕ ПР ИНИВИТИТЕТ НЕ ПР ИНИВИТЕТ НЕ ПР	INST IF PEDEFROF INST ID REDROBATA ALU Out 88888488 WB Data 88888888 WB Data 88888888 WB Data 88888888 WB DATA 888888888 INST IT PEDEFROF INST ID REDROBATA	X30: 15 09000000 PSIDATA PSID	x27: x31: rs20 rs2n PCdu D/C- PCIF nLUC WR- Rege CODE CODE CODE CODE FS20 PS20 PS20 PS20 PS20 PS20
个 word 计时器第 三拍 写回第一 个 word	ПР ИНИВИНА НЕ ПР ИНИВИНА НЕ ПР ИНИВИНИ НЕ ПР ИНИВИРИ НЕ ПР ИНИВИТИТЕТИТЕТИТЕТИТЕТИТЕТИТЕТИТЕТИТЕТИТЕТ	INST IF PEDEFROF INST ID REDROBATA ALU Out 88888488 WB Data 88888888 WB Data 88888888 CODE 81 8108883 CODE 85 81085283 CODE 89 80188923 CODE 80 18892383 INST ID REDROBATA INST ID REBRIARRE	X30: t5 09000000 PSIDATA BUBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB	x27: x31: rs20 rs2n PCJI nLUC WR- Regg CODE CODE CODE x31: rs2n PCJII pCJII pCJII pCJII pCJII pCJII pCJII
个 word 计时器第 三拍 写回第一 个 word 计时器第	ПЕ ИВИВИВНА ПЕ ПЕ ИВИВИВНА ПЕ ПЕ ИВИВИВНА ПЕ ПЕ ВВИВИВИЯ ЗА ПЕ ЦЕТИ ВВИВИВИ ЗВ ПЕТЕ ВВИВИВИ ЗВ ПЕТЕ ВВИВИВИ ЗВ ПЕТЕ ВВИВИВИ ЗВ ПЕТЕ ВВИВИВИ ЗВ ЗВ ПЕТЕ ВВИВИВИ ЗВ ЗВ ТЕТИ ВВИВИВИ ЗВ ЗВ ТЕТИ ВВИВИВИ ЗВ ПЕТЕ ВВИВИВИВИ ЗВ ПЕТЕ ВВИВИВИ ЗВ ПЕТЕ ВВИВИ ЗВ ВВ ПЕТЕ ВВИВИ ЗВ ВВ	INST IF PEDEFROF INST ID REDROBATA INST ID REDROBATA INST IN ALBREMARA INST IN ALBREMARA INST IN BERROBARA INST IN BERROBARA ALU Out 88888488 WB Addr 88888888 WB Addr 88888888 CODE 81 8108883 CODE 85 81085283 CODE 85 81085283 CODE 89 80188923 CODE 80 48892383 CODE 80 48892383 INST IN PEDEFROF INST ID REDROBATA INST IN REDROBATA INST IN BERROBARA IN	X30: 15 09800000 PSIDATA BUBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB	x27: x31: rs20 rs2n rs2n rcdu D/c- rc IE nLUC WR- Regg CODE CODE CODE x31: rs2n rs2n rs2n rs2n rs2n rs2n rs2n rs2n
个 word 计时器第 三拍 写回第一 个 word 计时器第	ПР ИНИВИНИ ПО ПРО ПРО ПРО ПРО ПРО ПРО НЕИВИНИ ЗВ ПРО	INST IF PEDEFROF INST ID REDROBATA ALU Out 88888488 WB Addr 88888888 CODE 81 8108883 CODE 85 81085283 CODE 89 80188923 CODE 89 80188923 CODE 80 18882383 INST ID REDROBATA INST ID REBROBABA INST ID REBROBABABA INST ID REBROBABABA INST ID REBROBABABA INST ID REBROBABABABA INST ID REBROBABABABABABABABABABABABABABABABABABAB	X30: t5 09000000 PSIDATA PSIDATA PSIDATA PSIDATA PSIDATA PSIDATA PSIDATA PSIDATA PSIDATA CPU-DA CPU-DA CODE-02 01C01103 CODE-04 01C01103 CODE-06 21002003 CODE-08 00101223 CODE-08 00101223 CODE-08 00101223 CODE-08 00101223 CODE-09 001000000000000000000000000000000000	x27: x31: rs20 rs20 rs20 rs20 rc1i nLUC WR- Regi CODE CODE CODE x31: rs20 rs20 rs20 rs20 rs20 rs20 rs20 rs20
个 word 计时器第 三拍 写回第一 个 word 计时器第	IF HAMBHAHA	INST IF PEDEFROR INST ID REDROBATA ALU Out 88888488 WB Addr 88889888 WB Addr 88889888 CODE-81 81C85283 CODE-85 81C85283 CODE-80 48892383 CODE-80 48892383 INST ID REDROBATA INST ID REBROBARA IN	X30: 15 00000000 PSIDATA PSIDATA PSIDATA PSIDATA PSIDATA PSIDATA PSIDATA PSIDATA PSIDATA CPU-DA CODE-02 01C01103 CODE-04 01C01103 CODE-06 21002003 CODE-07 00000000000000000000000000000000000	x27: x31: rs20 rs2n PCJ; nLUC WR- Regg, CODE CODE x31: rs2n rs2n rs2n rs2n rs2n rs2n rs2n rs2n
个 word 计时器第 三拍 写回第一 个 word 计时器第	IF HAMBHAHA IF IF HAMBHAHA IF IF IF IF IF IF IF I	INST IF PEDEFROF INST ID REDROBATA INST ID REDROBATA INST IN ALBREMARA INST IN ALBREMARA INST IN BERROBARA INST IN BERROBARA ALU Out 88888488 WB Addr 88888888 WB Addr 88888888 CODE 81 8108883 CODE 85 81085283 CODE 85 81085283 CODE 89 80188923 CODE 80 48892383 CODE 80 48892383 INST IN PEDEFROF INST ID REDROBATA INST IN REDROBATA INST IN BERROBARA IN	X30: t5 09000000 PSIDATA PSIDATA PSIDATA PSIDATA PSIDATA PSIDATA PSIDATA PSIDATA PSIDATA CPU-DA CPU-DA CODE-02 01C01103 CODE-04 01C01103 CODE-06 21002003 CODE-08 00101223 CODE-08 00101223 CODE-08 00101223 CODE-08 00101223 CODE-09 001000000000000000000000000000000000	x27: x31: rs20 rs20 rs20 rs20 rc1i nLUC WR- Regi CODE CODE CODE x31: rs20 rs20 rs20 rs20 rs20 rs20 rs20 rs20

后续重复三次写回 word 后,状态转移至读 word,后面情况和上述分析的 miss read 类似,这里不做赘述。

指令 14 访问 set1,line0,同指令一一样,但是指令一是读取指令,没有写,因此 cache 是非 dirty 的,不需要写回。

四、讨论、心得

本实验只需要填几个空完整有限状态机的状态转移过程,因此整体实验比较简单,过程中也没有遇到什么困难。

另外,跑仿真时发现只有前面几条指令的波形,在设置中修改 simulation 的时间后解决。