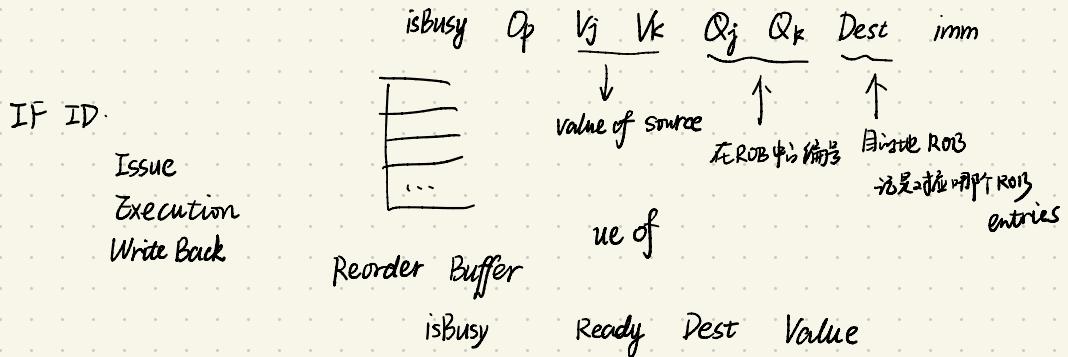


Tomasulo 算法: Reservation Station 16条 32条?



Hazards

RS: ① 时序部分: 上升沿时寻找可执行的语句 ( $Q_j = Q_k \Rightarrow$ )

发送至 EX module

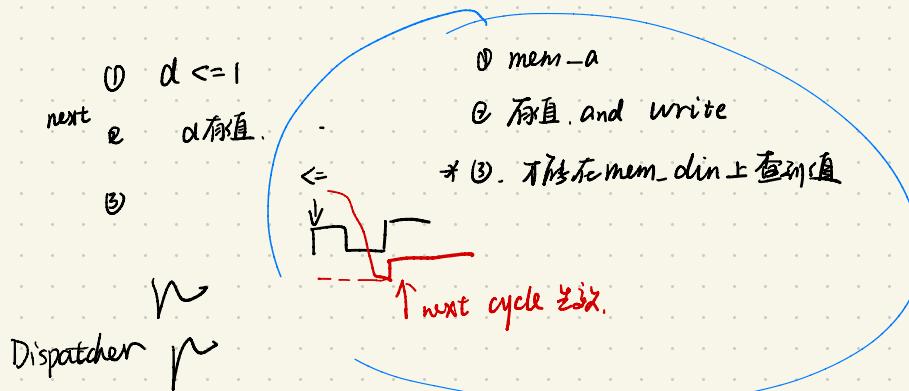
EX 内组合逻辑

Load Buffer?

数据相关性

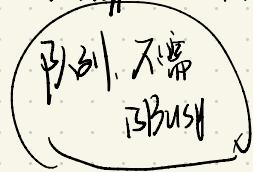
② 时序部分: 根据 EX 的结果更新  $V_j$ ,  $V_k$

ROB: 顺序推送 Commit



去 Reg 中 查是否处于 random ing 状态

Reorder Buffer: is Ready. Op. dest Reg. value to\_pc



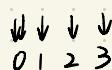
pc is calculated

ROB

分析老ROB行为: 二 左开右闭 head  
tail  
tail+1 == head



full:



head

0

head == tail

BP溢出 / 满

Commit	LOAD	→ TO REG	head
	STORE	→ TO LSB · begin store	
	BRANCH	↓	
	JUMP	→ PC (IF)	
	CALL	→ Reg	

LSB, 4字节 STORE 队首 → ROB. Ready

ROB commit 任务队列

甘执行完后再次通知 ROB 队列

(或退队, 但在跳转时候不能停止 STORE  
(包括 ready))

LSB. Mem - ctrl.

LSB:

isBusy Op Vj Vk Qj Qk destRw  
imm.

LSB

Today's Topic: Mem - ctrl.

deal with I-Cache.

drop 支持

每 1 CLK 只能执行一件事

这种向内存写

必须得至少间隔一个 cycle

与内存需要一个 cycle

状态机的逻辑



Todo: Mem - Ctrl.

所有排队逻辑处理

Load. Store

同一时间会有一个

/IDLE

Do not write

LSB LSB

① LOAD.

① check Mem - ctrl 逻辑结果?

返回队列和 ROB + 译码队

否则 继续推

STORE.

① check ROB 是否命中 [译码队]

命中

看 Mem - ctrl 逻辑结果?

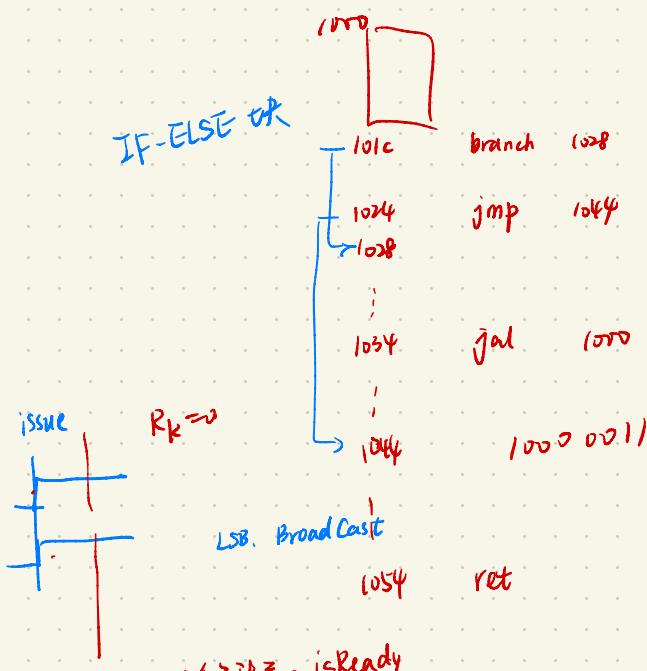
返回队列

译码队

否则 继续推

不命中

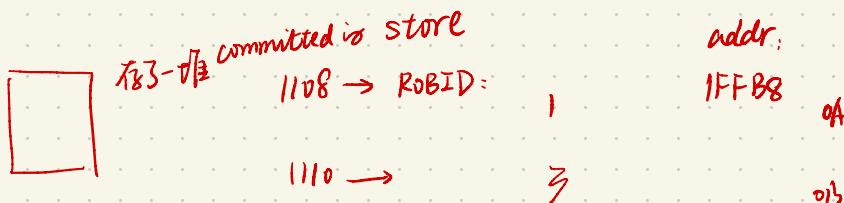
看 ROB



ROB 为空  $\Rightarrow$  isReady

Solution: issue  $\rightarrow$  RS/ LSB PT

Broadcast PT 为空  $\Rightarrow$  issue to VJ (Dj)



这个 clock. 1108. 但是 jump-flag



①

有多个 store!

第二个问

input 测试

This week tasks

pass sim and test tak.c / pi.c / heart.c  
on fpga board.

模块级顶层: 通过 ready-in 处理  
input 处理 (to do)

~~array-test1.c~~

~~array-test2.c~~

~~expr.c~~

~~looper.c~~ WA

~~gcd.c~~

~~Walne2.c~~

~~multidarray.c~~

~~hanoi.c~~

~~magic.c~~

~~queens.c~~ WA

~~psort.c~~

~~bulgarian.c~~ 卡死

~~basicopti.c~~

~~manymemory.c~~

~~statement-test.c~~

~~superloop.c~~

~~uartbootm.c~~

step 49 LSB full

bug: LSB. last-committed 位移  
不能通过比较, 应+1 取模

bug: isSendToROB[i]. 每次触发一次

考虑

13-1

$c=3$

3,5

$$5 \times 8 - 1 - 3 = 9$$

$\angle 2^{\circ}$

看起來 store 在 3

$[1:0]$

$\angle 2^{\circ}$

記得

$$S3 = 5008$$



第4次  
那次的值  
不存

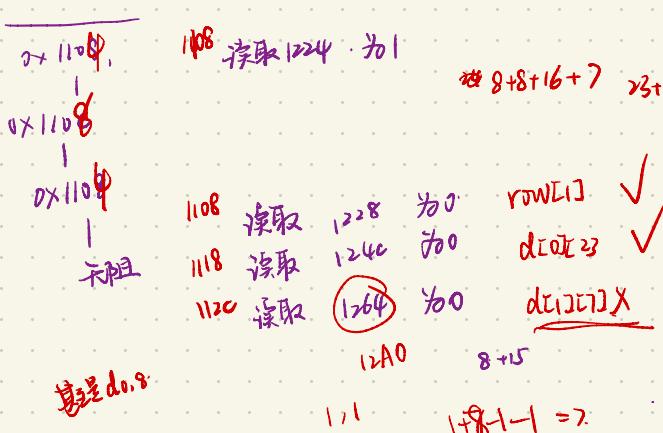
$$+6 \times 4 + 7 \times 4 = 23 \times 4 \rightarrow 91$$

$$0+8-1 \rightarrow$$

last	store	1178	存到	12A0	1	$d[1:0]$
		1170	存到	1244	1	$d[0:0]$
		1180	存到	1224	1	row[2] addr.
		1184	存到	1204	0	col[0] addr.

$\angle 0111012$

二叉入DFS



$S3: 1204$

$S2: 1244 \rightarrow 1248$

$S1: 1224 \rightarrow 1228$

$S0: r \rightarrow N-1$

1204

$\uparrow S3 + A5 \rightarrow$

$64 + A5$

要加 39 -21

$$A5 = \frac{r}{N} + A4 \quad N$$

$$A5 = A5 - S4 \quad \frac{r}{N} + A4 - C$$

$$A5 = A5 + A5 \quad \text{wrong}$$

$$A5 = A5 - X4$$

$$+64/4 = 16$$

$$r + N - 1 - C \quad 158$$

$$156 = 64 + 92 = 64 + 4 \times 23$$

line 35066 time 340715

3396

1108: a5>0

robID  
03.  
04  
06  
07

110C: a5=9  
1110: a5=8  
1118: a5=23  
110C: a5=92  
32?

实测BA: a5<0

Correct

113C  
1184  
认为 b[1] [r+N+1-c] == 0

113C

1140

$B_p b[1] [r+N+1-c] == 1$

r 从下→左循环

开始认为这是个判断语句

1308

s3. col: 1390

s3: 5008

最后一个判断

0 1 0  
1 3 0  
2 5 0  
3 7 0  
4 2 0  
5 0 0  
6 6 0  
7 4 0  
0 0 0  
0 0 0

1120, slli a5, a5, 0x2

1130, add a5, s3, a5

34273: 1130 X

333129 ns

$5008 + 88 = 5096$

ROBID 用尽为0.

but 这里

被塞了

I Cache:

IsWaitMem

IsWaitingMem

If hit

line 8x5  
time 8x79



ROBID=2

issue: dest ROB = 1

Bug: ready 3 但因为 & = 0

5 7倍. 被掩盖



$S2 = 1FFD4$

$SD = 1FFD8$

要修改, 11A8 → 不修改

To do:

addi \$0, \$0, 1

4

addi \$0, \$0, 1

0 1.

~~H~~

1 2 3 4  
5 6 7 8

head

↑ — — —

Issue

Send to ALM

Tell ROD

could commit

Commit Valid

Bug, Next full

To do: ① rdy-in 处理

② I/O Buffer full (for output)

③ Input is 特殊处理 (wait until it become the head of ROB)

④ 分支预测

rdy: 是否正在工作 · low: Cannot do anything!

— +

+ |

— +

i/o-buffer-full

不是 next-full

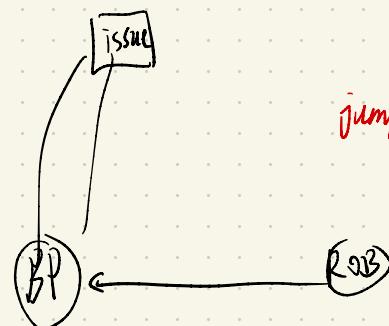
如果是分支 预测时

无论如何 ROB 知道 跳转了没有

预测时

jump 与否 先提前告诉 RS

↪ 0x1



1084

108L

so ← 8

1FFC9

ToPc ← +

① 去 Reg 查 ROBID (成功无)

有 → 看这次 commit 什么情况

② 否则、再看看 ROB 中真做了没有