Architecture

庄永昊

2019 年 12 月 19 日

1 Out of Order Execution

Problem A

Inst	Enter ROB	Issue	WB	Commit	OP	Dest	Src1	Src2
I_1	-1	0	3	4	FMUL.D	r0	f0	f1
I_2	0	4	9	10	FADD.D	r1	f2	r0

Problem B

Inst	Enter ROB	Issue	WB	Commit	OP	Dest	Src1	Src2
I_1	-1	0	3	4	FMUL.D	r0	f0	f1
I_2	0	4	9	10	FADD.D	r1	f2	r0
I_3	5	10	15	16	FADD.D	r0	r1	f0

Problem C

Inst	Enter ROB	Issue	WB	Commit	OP	Dest	Src1	Src2
I_1	-1	0	2	3	FLD	r0	x1	
I_2	0	1	3	4	FLD	r1	x1	
I_3	1	4	7	8	FMUL.D	r2	r0	r1
I_4	2	8	13	14	FADD.D	r3	f2	r2
I_5	4	5	8	9	FLD	r0	x1	
I_6	5	9	14	15	FADD.D	r1	r3	r0

2 Virtual Memory&Alias Problem

Problem A

There are $2^32/4096 = 2^20$ pages in need, and each page of page table can contain $4096/4 = 1024$ enteries, so the number of level is $20/\log_2(1024) = 2$.

Problem B

The aliasing promlem comes up when a cache is virtually mapped and a couple of virtual addresses are mapped to the same physical address. The solution is effective when $k \geqslant L + b$.

This solution is effective because:

If two virtual addresses are mapped to the same physical address, they share the same page offset and thus the same cache index. And as the cache is direct-mapped, the two virtual addresses get the same tag, which is the physical tag they share.

Problem C

The page offset has $2 + \log_2 4096 = 14$ bits, the cache index has $\log_2(256) = 8$ bits and the cache offset is $\log_2(64) + 2 = 8$ bits, so it needs to be $2^{8+8-14} = 4$ way associative.

3 Branch Prediction

Problem A

i	Instruction	Counter	Prediction	Actual
i=0	blt x2,x5,skip 1	01	not taken	taken
	bge x2,x5 skip 2	01	not taken	not taken
	bnez x4,loop	01	not taken	taken
i=1	blt x2,x5,skip 1	10	taken	not taken
	bge x2,x5 skip 2	00	not taken	taken
	bnez x4,loop	10	taken	taken
i=2	blt x2,x5,skip 1	01	not taken	taken
	bge x2,x5 skip 2	01	not taken	not taken
	bnez x4,loop	11	taken	taken
i=3	blt x2,x5,skip 1	10	taken	not taken
	bge x2,x5 skip 2	00	not taken	taken
	bnez x4,loop	11	taken	not taken

i	Instruction	Global	Counter 0	Counter 1	Prediction	Actual
i=0	blt x2,x5,skip 1	0	01	01	not taken	taken
	bge x2,x5 skip 2	1	01	01	not taken	not taken
	bnez x4,loop	0	01	01	not taken	taken
i=1	blt x2,x5,skip 1	1	10	01		not taken
	bge x2,x5 skip 2	0	01	00	not taken	taken
	bnez x4,loop	1	10	01	taken	taken
i=2	blt x2,x5,skip 1	1	10	00	not taken	taken
	bge x2,x5 skip 2	1	10	00	not taken	not taken
	bnez x4,loop	0	10	10	taken	taken
i=3	blt x2,x5,skip 1	1	10	01	not taken	not taken
	bge x2,x5 skip 2	0	10	00	taken	taken
	bnez x4,loop	1	11	10	taken	not taken

Branch	Accuracy
blt	2/4
bge	3/4
bnez	1/4
overall	6/12

Problem B

Problem C

bge will, because it follows a blt that usually goes to the opposite of the bge. (the one less than 5 is always not greater than or equal to 4)

Problem D

BTB can be placed in fetch stage for a quick prediction, but it cannot cover lots of branches.

Oppositely, BHT in decode is large enough to have a number of entries and give a precise prediction, but it will introduce bubbles since it is in decode stage.