Today

- Learning outcomes
 - Describe what a processor does when we makes an incorrect prediction
- How we'll get there
 - Review how/when a mis-predict happens.
 - Practice rearranging loops to improve prediction behavior.

Recall

- To mitigate control hazards due to conditional branches, we said that we could make a guess (e.g., prediction) and then keep executing based on that prediction.
- But: What do we do if that prediction is wrong?

Mis-Prediction

- What happens if you are wrong?
- Consider:

irmovq 0x1, %rax
irmovq 0x1, %rbx
subq %rax, %rbx
je skip
irmovq 0x5000, %rcx
irmovq 0x5000, %rdx
<more instructions here>
skip:
addq %rcx, %rbx

mulg %rdx, %rax

Let's assume that our prediction algorithm is "never taken"

You have two bad instructions in the pipeline!

- That is, instructions you should not have executed!
- So what do you do????

You cancel or squash or quash them!

So:

- In the best case, conditional branches incur no penalty
- In the worst case (where you mispredict), it's no worse than if you'd done no prediction)

Branch Mis-Predicts (1)

R[%rcx] = 1, $R[%rdx] = 1$
R[%rax] = 10, R[%rbx] = -10

addq %rax, %rbx

je Skip

irmovq 0x5000, %rcx irmovq 0x5000, %rdx

addq %rax, %rbx	Decode	Execute	Memory	Writeback	Skip: addq %rcx, %rbx			
	Fetch	Decode	Execute	Memory	Writeback	mulq %rdx, %rax		
		Fetch	Decode	Execute	Memory	Writeback		
			Fetch	Decode	Execute	Memory	Writeback	
				Fetch	Decode	Execute	Memory	

Branch Mis-Predicts (2)

addq %rax, %rbx	valA=R[%rax]=10 valB=R[%rbx]=-10	Execute	Memory	Writeback		Skip: addq %rcx, %rbx mulq %rdx, %rax Writeback		
	je skip	Decode	Execute	Memory	Writeback			
		Fetch	Decode	Execute	Memory	Writeback		
			Fetch	Decode	Execute	Memory	Writeback	
				Fetch	Decode	Execute	Memory	

R[%rcx] = 1, R[%rdx] = 1 R[%rax] = 10, R[%rbx] = -10

je Skip

addq %rax, %rbx

irmovq 0x5000, %rcx irmovq 0x5000, %rdx

Branch Mis-Predicts (3)

irmovq 0x5000, %rdx Skip: valA=R[%rax]=10 valE = valA+valB=0 addg %rax, %rbx Writeback Memory addq %rcx, %rbx ZF = 1 valB=R[%rbx]=-10 mulq %rdx, %rax nextPC =valP Writeback je skip Execute Memory Writeback Decode Memory Execute irmovq 0x5000, %rcx Writeback Decode Execute Memory Fetch Assume we predict NOT TAKEN Decode Memory Fetch Execute

R[%rcx] = 1, R[%rdx] = 1R[%rax] = 10, R[%rbx] = -10

je Skip

addq %rax, %rbx

irmovq 0x5000, %rcx

Fetch next instruction

R[%rax] = 10, R[%rbx] = -10

addq %rax, %rbx

R[%rcx] = 1, R[%rdx] = 1

je Skip

irmovq 0x5000, %rcx

irmovq 0x5000, %rdx

Branch Mis-Predicts (4)

addq %rax, %rbx	valA=R[%rax]=10 valB=R[%rbx]=-10	valE = valA+valB=0 ZF = 1		Writeback	Skip: addq %rcx, %rbx			
	je skip	valP	Cnd(0,'e')	Memory	Writeback	mulq %rdx, %rax		
		irmovq 0x5000, %rcx	valC = 0x5000	Execute	Memory Writeback			
irm		irmovq 0x5000, %rdx	Decode	Execute	Memory	Writeback		
				Fetch	Decode	Execute	Memory	

Uh oh! We can now evaluate the condition, and it turns out we should have jumped!

Branch Mis-Predicts (5)

					irmovq 0x5000, %rdx			
addq %rax, %rbx	valA=R[%rax]=10 valB=R[%rbx]=-10	valE = valA+valB=0 ZF = 1		R[%rbx] = 0			addq %rcx, %rbx	
	je skip	valC=skip	Cnd(0,'e')		Writeback	mulq %rdx	x, %rax	
		irmovq 0x5000, %rcx	valC = 0x5000	valE = 0x5000	Memory	Writeback		
			irmovq 0x5000, %rdx	valC = 0x5000	Execute	Memory	Writeback	
7				addq %rcx, %rbx	Decode	Execute	Memory	

R[%rcx] = 1, R[%rdx] = 1 R[%rax] = 10, R[%rbx] = -10

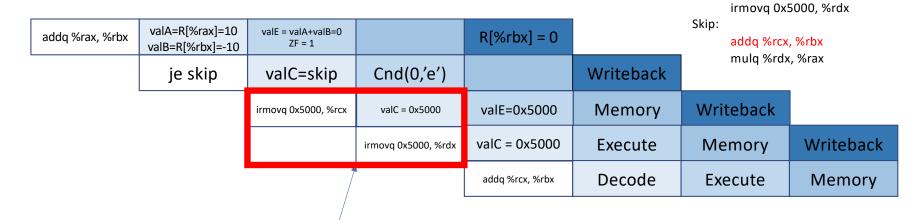
je Skip

addq %rax, %rbx

irmovq 0x5000, %rcx

OK – so now we are executing at the place we should have been executing after the jump

Branch Mis-Predicts (6)



But what about these instructions???

Bad news: You have started executing two instructions that you should not have.

R[%rcx] = 1, R[%rdx] = 1R[%rax] = 10, R[%rbx] = 0

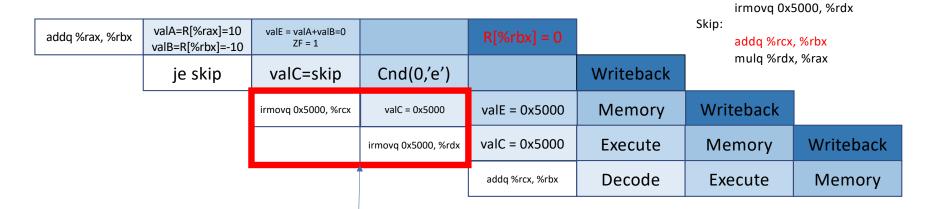
je Skip

addq %rax, %rbx

irmovq 0x5000, %rcx

Good news: They haven't modified any state that anyone has "seen."

Branch Mis-Predicts (7)



We will cancel/squash/quash them!
Make sure that no visible changes get made:

- No writeback
- No writes to memory
- No update of condition codes

Bad news: You have started executing two instructions that you should not have.

R[%rcx] = 1, R[%rdx] = 1R[%rax] = 10, R[%rbx] = -9

je Skip

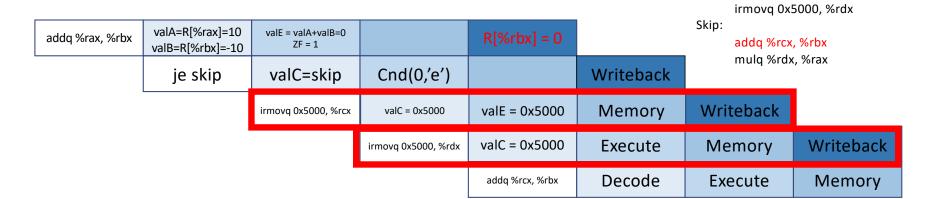
addq %rax, %rbx

irmovq 0x5000, %rcx

Good news: They haven't modified any state that anyone has "seen."

CPSC 313 10

Branch Mis-Predicts (8)



R[%rcx] = 1, R[%rdx] = 1R[%rax] = 10, R[%rbx] = -9

je Skip

addq %rax, %rbx

irmovq 0x5000, %rcx

+1

9 cycles/3 instructions completed = 3 CPI

CPSC 313 11