Example 5

Replication towards return or fall-through

#	without replication	favor returns	favor loops

1	L83
	NZ=B[a[2]]?0;
	PC=NZ==0,L02;
		NZ=B[a[2]]?0;
		PC=NZ==0,L02;
			NZ=B[a[2]]?0;
			PC=NZ==0,L02;

2	b[0]=B[a[2]++];
	NZ=b[0]?99;
	PC=NZ==0,L90;
		L09
		b[0]=B[a[2]++];
		NZ=b[0]?99;
		PC=NZ==0,L90;
			L09
			b[0]=B[a[2]++];
			NZ=b[0]?99;
			PC=NZ==0,L90;

	...	...	L08 ...

3		NZ=B[a[2]]?0;
		PC=NZ!=0,L09;
			NZ=B[a[2]]?0;
			PC=NZ==0,L02;

4	PC=L83;	a[6]=UK;
		PC=RT;
			b[0]=B[a[2]++];
			NZ=b[0]?99;
			PC=NZ!=0,L08;

5	L90 ...	L90 ...	L90 ...

6	L02
	a[6]=UK;
	PC=RT;
		L02
		a[6]=UK;
		PC=RT;
			L02
			a[6]=UK;
			PC=RT;

Measurements (cont.)

Percent of Instructions that are Unconditional Branches

		static			dynamic		
		SIMPLE	LOOPS	JUMPS	SIMPLE	LOOPS	JUMPS
Sun SPARC	average std.deviation	3.74% 1.78%	2.40% 1.99%	0.03% 0.12%	3.28% 2.71%	1.89% 2.56%	0.10% 0.30%
Motorola 68020	average std.deviation	5.08% 2.49%	3.42% 2.83%	0.04% 0.15%	4.14% 3.48%	2.47% 3.36%	0.13% 0.43%

Measurements (cont.)

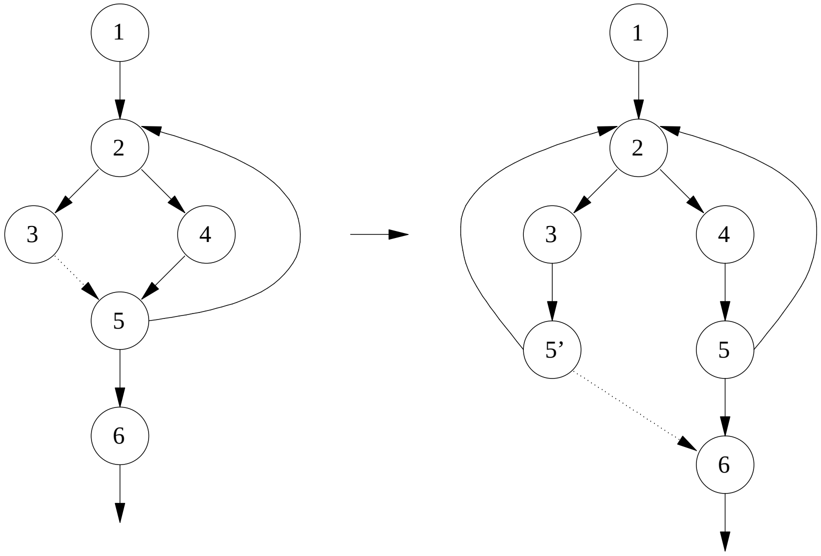
Percent Change in Miss Ratio and Instruction Fetch Cost for Direct-Mapped Caches

	cache size	1Kb		2Kb		4Kb		8Kb	
processor	context sw.	LOOPS	JUMPS	LOOPS	JUMPS	LOOPS	JUMPS	LOOPS	JUMPS
		Cache Miss Ratio							
Sun	on	-0.05%	+1.07%	-0.22%	-0.07%	+0.03%	+0.25%	+0.01%	+0.11%
SPARC	off	-0.03%	+1.07%	-0.22%	-0.08%	+0.03%	+0.21%	+0.01%	+0.07%
Motorola	on	+0.08%	+1.26%	+0.04%	+0.75%	+0.01%	+0.09%	+0.01%	+0.07%
68020	off	+0.08%	+1.25%	+0.03%	+0.70%	+0.01%	+0.05%	+0.01%	+0.03%
		Instruction Fetch Cost							
Sun	on	-2.73%	+3.44%	-3.80%	-5.24%	-2.26%	-2.94%	-2.40%	-3.98%
SPARC	off	-2.64%	+3.68%	-3.87%	-5.33%	-2.24%	-3.13%	-2.47%	-4.30%
Motorola	on	-3.07%	+1.69%	-3.26%	-0.63%	-3.58%	-5.13%	-3.57%	-5.30%
68020	off	-3.04%	+1.86%	-3.28%	-0.71%	-3.61%	-5.48%	-3.60%	-5.66%

*fetch cost = cache hits * cache access time + cache misses * miss penalty*

Moving Unconditional Jumps out of Loops

jump could be moved to the loop exit when it can’t be avoided


1
2
3
4
5
6
1
2
3
4
5’
5
6