

# Perfect Branch Prediction vs. Perfect Data Cache

## Assumptions

- Single issue out-of-order processor, unlimited window size
  - 10 cycle branch delay
  - Single cycle access, nonblocking cache, 1 word block size
  - Memory accesses have 10% miss rate
  - 100 cycle main memory access
  - No structural hazards
- 

```
for (i=0; i < 100,000; i++)
    Y(i) = (X(i) == 0.0) ? X(i) : Y(i-1);

    addiu $s3, $s0, #400000      ; initialize $s3
loop: lw $f2, 0($s0)             ; load X(i)
    c.eq.s $f0, $f2              ; X(i) == 0.0
    bfpf yi-1                    ; 50% mispredict
    j yi                          ;
    yi-1: lw $f2, -4($s1)         ; Y(i-1)
    yi: sw $f2, 0($s1)           ; store Y(i)
    addiu $s0, $s0, #4           ; increment X index
    addiu $s1, $s1, #4           ; increment Y index
    slt $s2, $s0, $s3            ; test if done
    bnez $s2, loop               ; 0% mispredict
```

Which performs better perfect branch prediction or perfect cache?