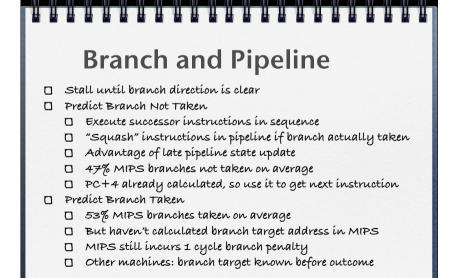


Software Scheduling □ Try producing fast code for a = b + c; Fast code: d = e - fLW Rb.b assuming a, b, c, d, e, and fin memory. W Rc, c Slow code: LW Re,e LW Rb.b ADD RA, Rb, RC LW RC,C ADD Ra, Rb, RC LW Rf,f SW a,Ra SW a, Ra LW Re,e SUB Rd, Re, Rf LW Rf,f SW d,Rd SUB Rd, Re, Rf SW d,Rd

Stall and Performance If CPI = 1, 30% branch, Stall 3 cycles => new CPI = ?



Branch and Pipeline

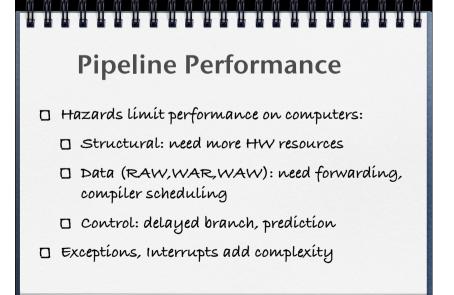
- O Delayed Branch
 - Define branch to take place AFTER a following instruction

branch instruction sequential successor₁ sequential successor₂

sequential successorn

branch target if taken

- 1 slot delay allows proper decision and branch target address in 5 stage pipeline
- MIPS uses this



Branch & Pipeline

Assume 4% unconditional branch, 6% conditional branch- untaken, 10% conditional branch-taken

Scheduling scheme	Branch penalty	CPI	speedup v. unpipelined	speedup v. stall
Stall pipeline	3	1.60	3.1	1.0
Predict taken	1	1.20	4.2	1.33
Predict not taker	1 1	1.14	4.4	1.40
Delayed branch	0.5	1.10	4.5	1.45

Pipeline speedup =
$$\frac{\text{Pipeline depth}}{1 + \text{Branch frequency} \times \text{Branch penalty}}$$

Pipeline Performance

☐ Speed up Pipeline Depth; if ideal CPI is 1, then:

Speedup =
$$\frac{\text{Pipeline depth}}{1 + \text{Pipeline stall CPI}} \times \frac{\text{Cycle Time}_{\text{unpipelined}}}{\text{Cycle Time}_{\text{pipelined}}}$$