

Ripasso concetti base: Pipeline, forwarding unit, branch prediction

Andrea Bartolini <a.bartolini@unibo.it>

(Architettura dei) Calcolatori Elettronici, 2023/2024

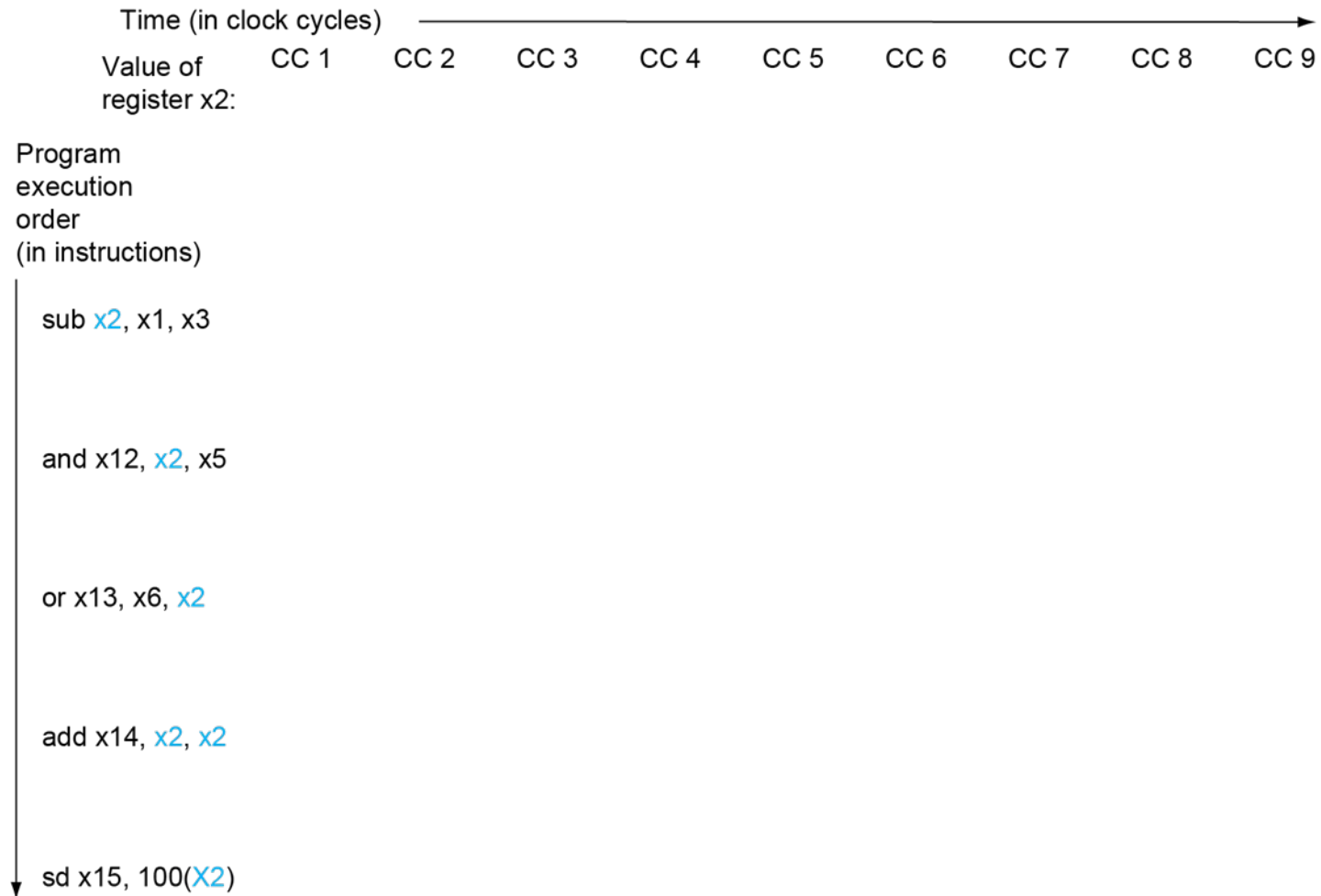
Data Hazards in ALU Instructions

- Consider this sequence:

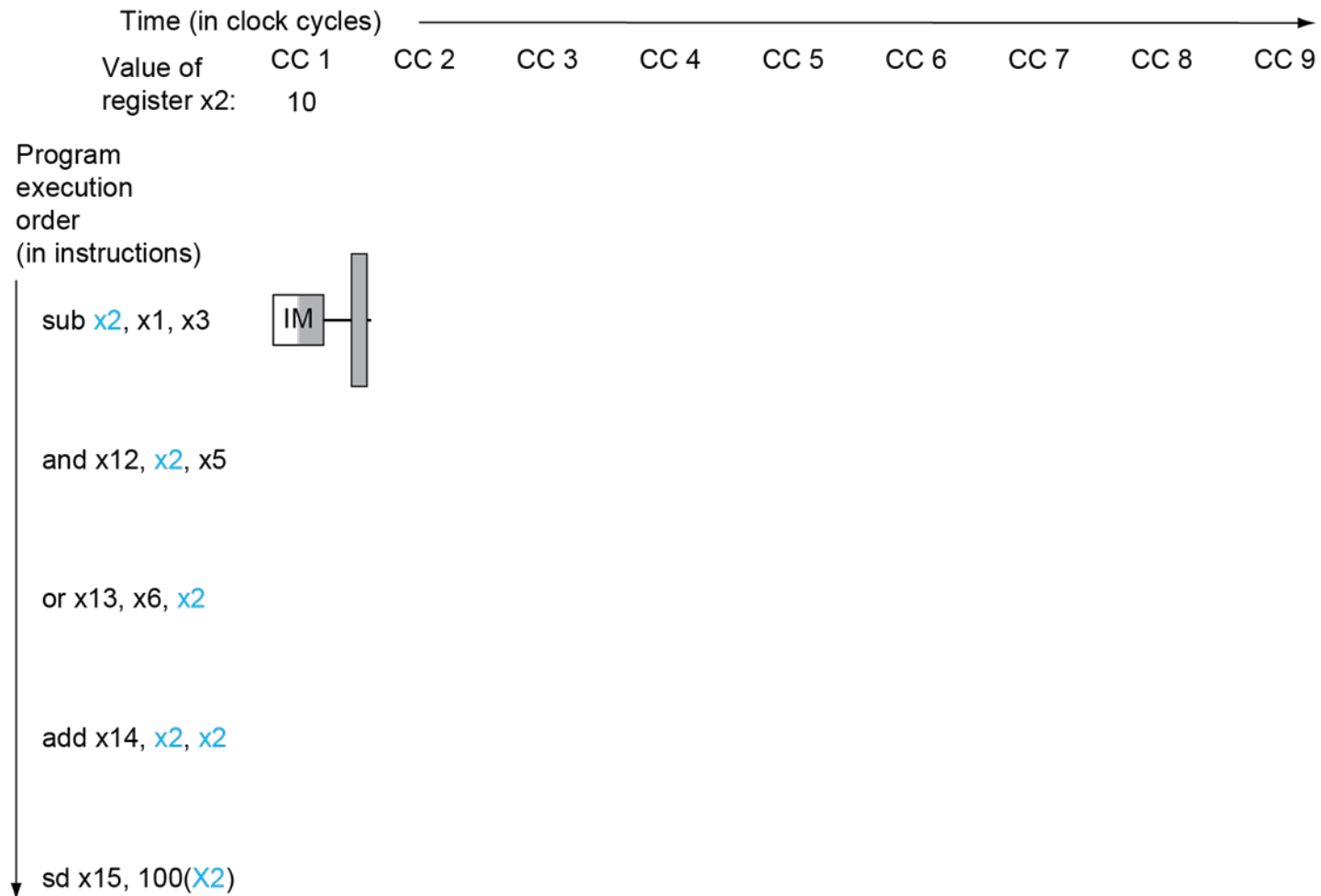
```
sub  x2, x1, x3
and  x12, x2, x5
or   x13, x6, x2
add  x14, x2, x2
sd   x15, 100(x2)
```

- We can resolve hazards with forwarding
 - How do we detect when to forward?

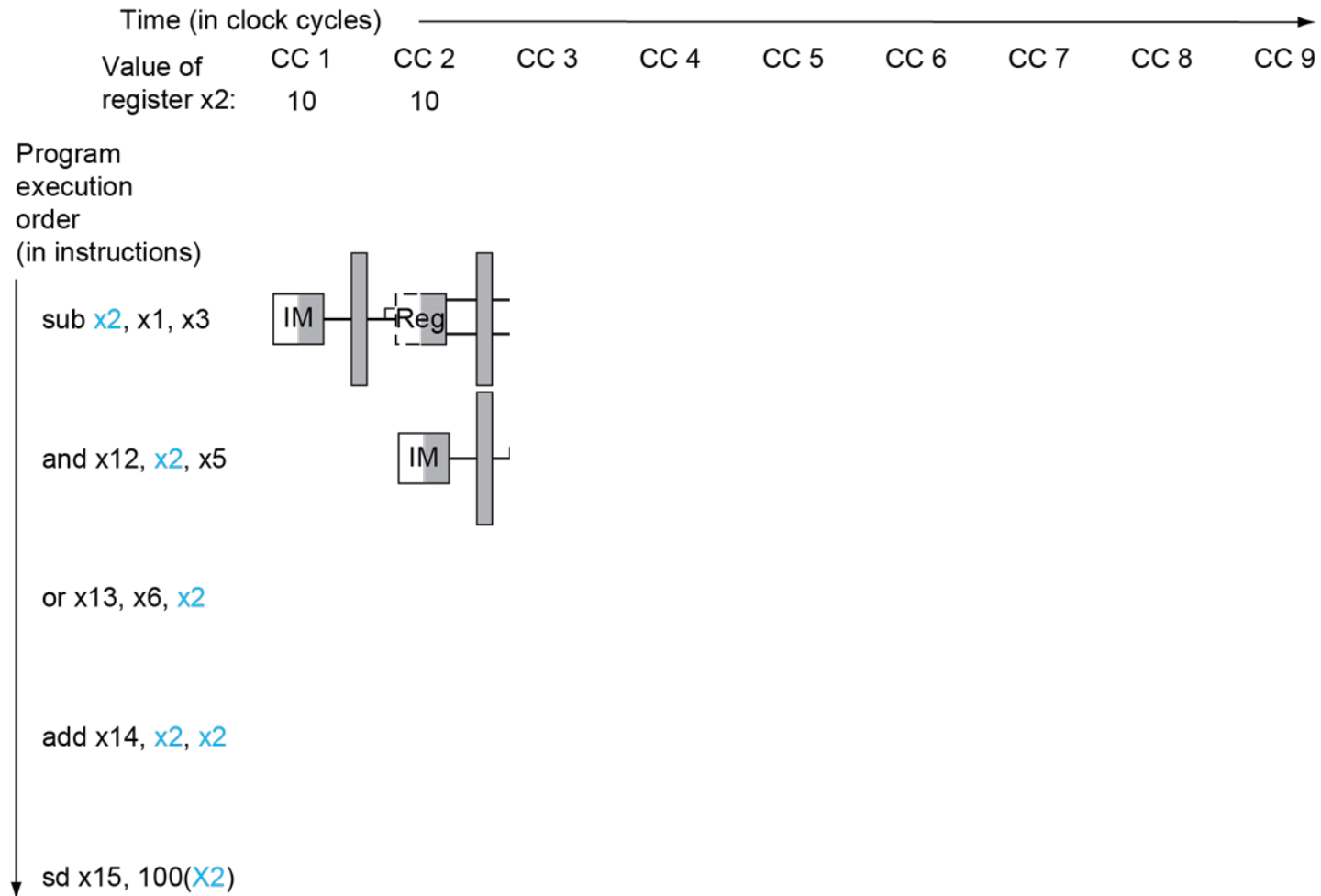
Dependencies & Forwarding



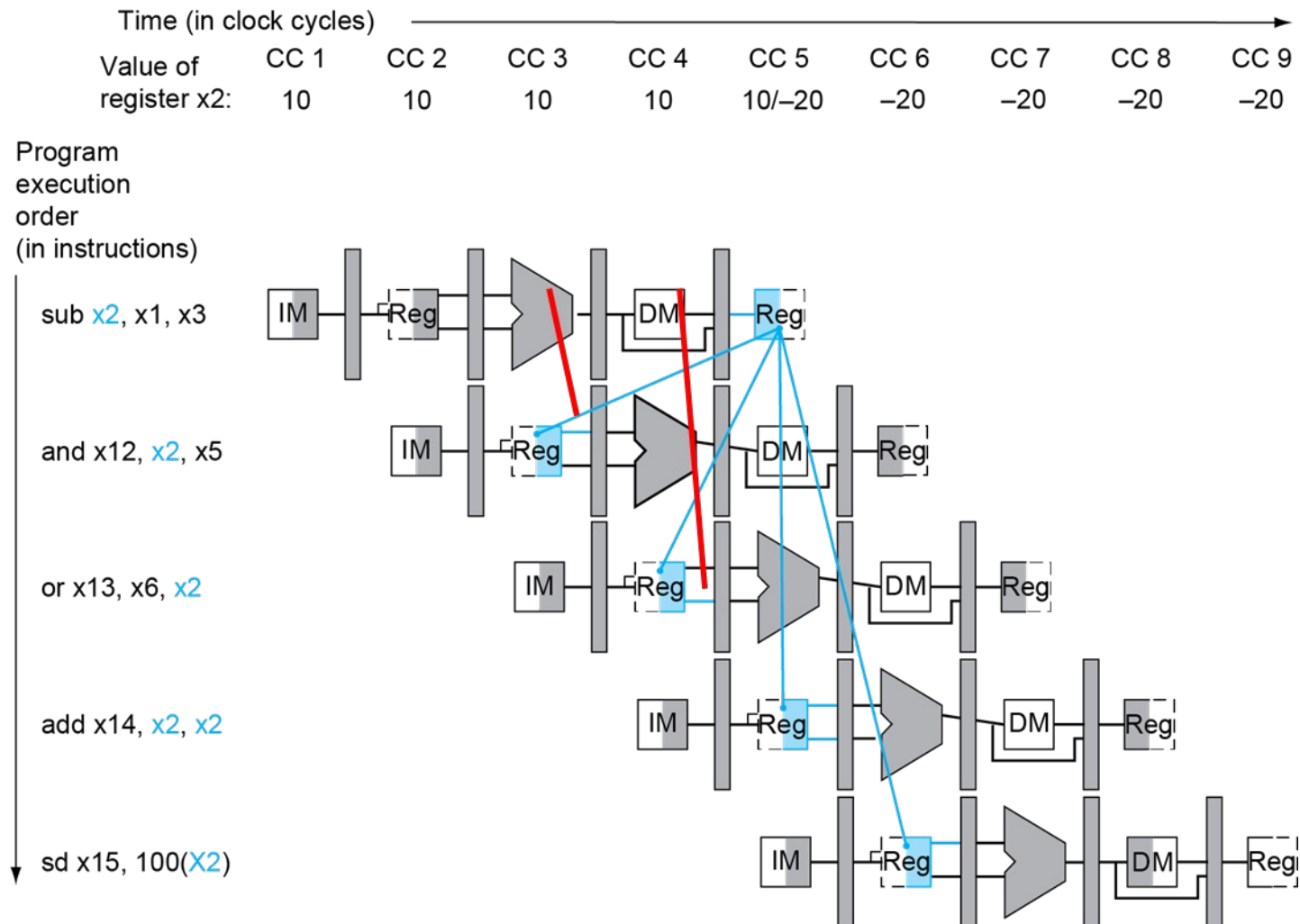
Dependencies & Forwarding



Dependencies & Forwarding

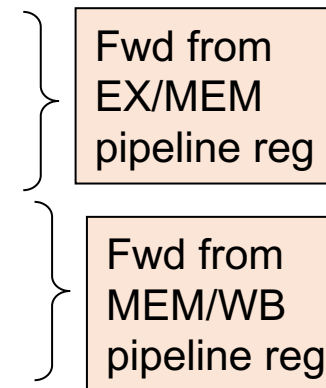


Dependencies & Forwarding



Detecting the Need to Forward

- Pass register numbers along pipeline
 - e.g., ID/EX.RegisterRs1 = register number for Rs1 sitting in ID/EX pipeline register
- ALU operand register numbers in EX stage are given by
 - ID/EX.RegisterRs1, ID/EX.RegisterRs2
- Data hazards when
 - 1a. EX/MEM.RegisterRd == ID/EX.RegisterRs1
 - 1b. EX/MEM.RegisterRd == ID/EX.RegisterRs2
 - 2a. MEM/WB.RegisterRd == ID/EX.RegisterRs1
 - 2b. MEM/WB.RegisterRd == ID/EX.RegisterRs2



Detecting the Need to Forward

- But only if forwarding instruction will write to a register!
 - EX/MEM.RegWrite, MEM/WB.RegWrite
- And only if Rd for that instruction is not x0
 - EX/MEM.RegisterRd \neq 0,
MEM/WB.RegisterRd \neq 0

Double Data Hazard

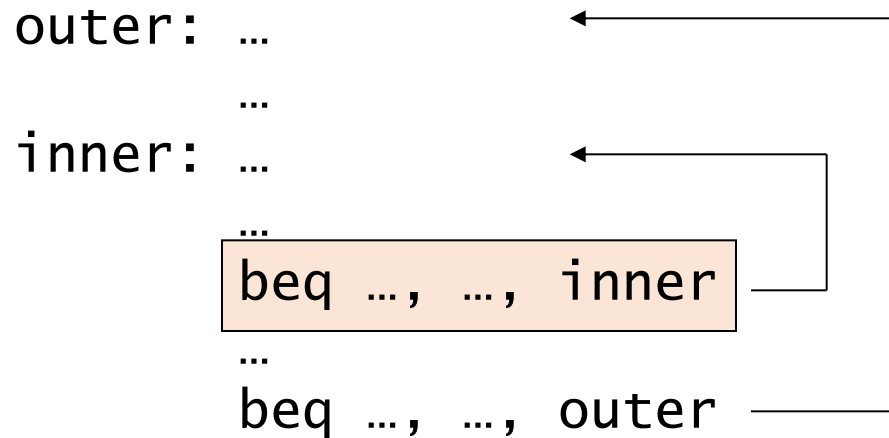
- Consider the sequence:
 add x1, x1, x2
 add x1, x1, x3
 add x1, x1, x4
- Both hazards occur
 - Want to use the most recent
- Revise MEM hazard condition
 - Only fwd if EX hazard condition isn't true

Dynamic Branch Prediction

- In deeper and superscalar pipelines, branch penalty is more significant
- Use dynamic prediction
 - Branch prediction buffer (aka branch history table)
 - Indexed by recent branch instruction addresses
 - Stores outcome (taken/not taken)
 - To execute a branch
 - Check table, expect the same outcome
 - Start fetching from fall-through or target
 - If wrong, flush pipeline and flip prediction

1-Bit Predictor: Shortcoming

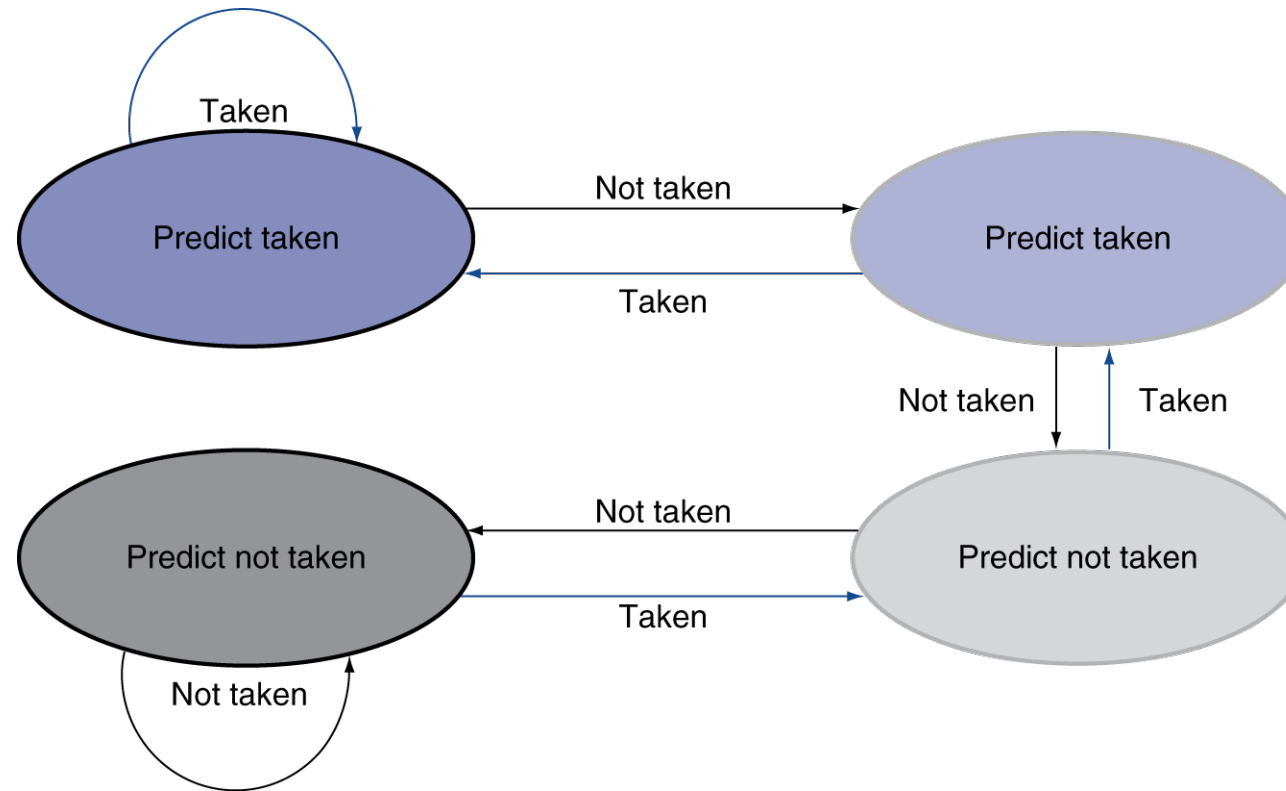
- Inner loop branches mispredicted twice!



- Mispredict as taken on last iteration of inner loop
- Then mispredict as not taken on first iteration of inner loop next time around

2-Bit Predictor

- Only change prediction on two successive mispredictions



Calculating the Branch Target

- Even with predictor, still need to calculate the target address
 - 1-cycle penalty for a taken branch if address calculation not done in ID.
- Branch target buffer
 - Cache of target addresses
 - Indexed by PC when instruction fetched
 - If hit and instruction is branch predicted taken, can fetch target immediately