

250P: Computer Systems Architecture

Lecture 3: Basics of pipelining

Anton Burtsev
January, 2019

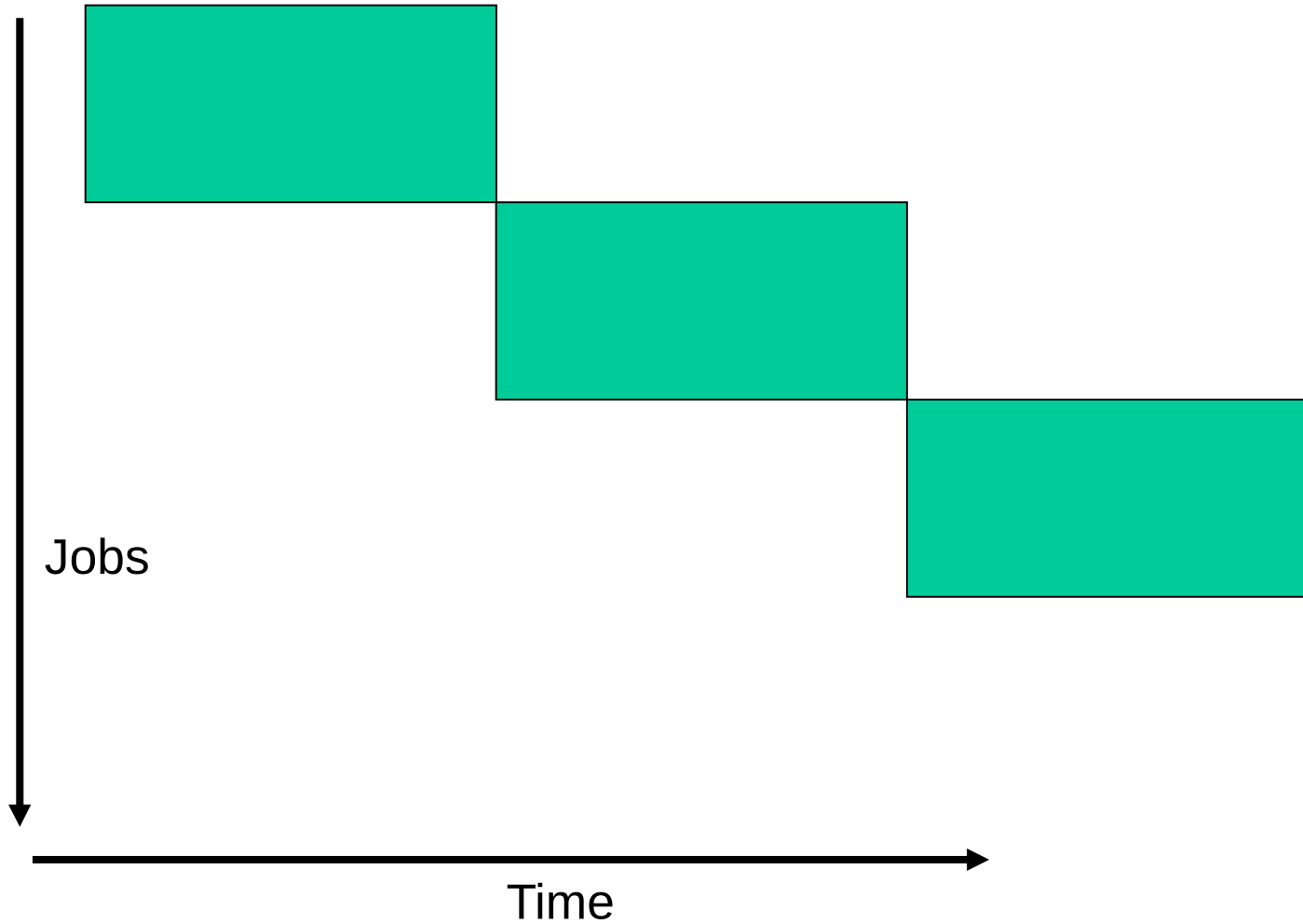
Building a Car



Building a Car

Unpipelined

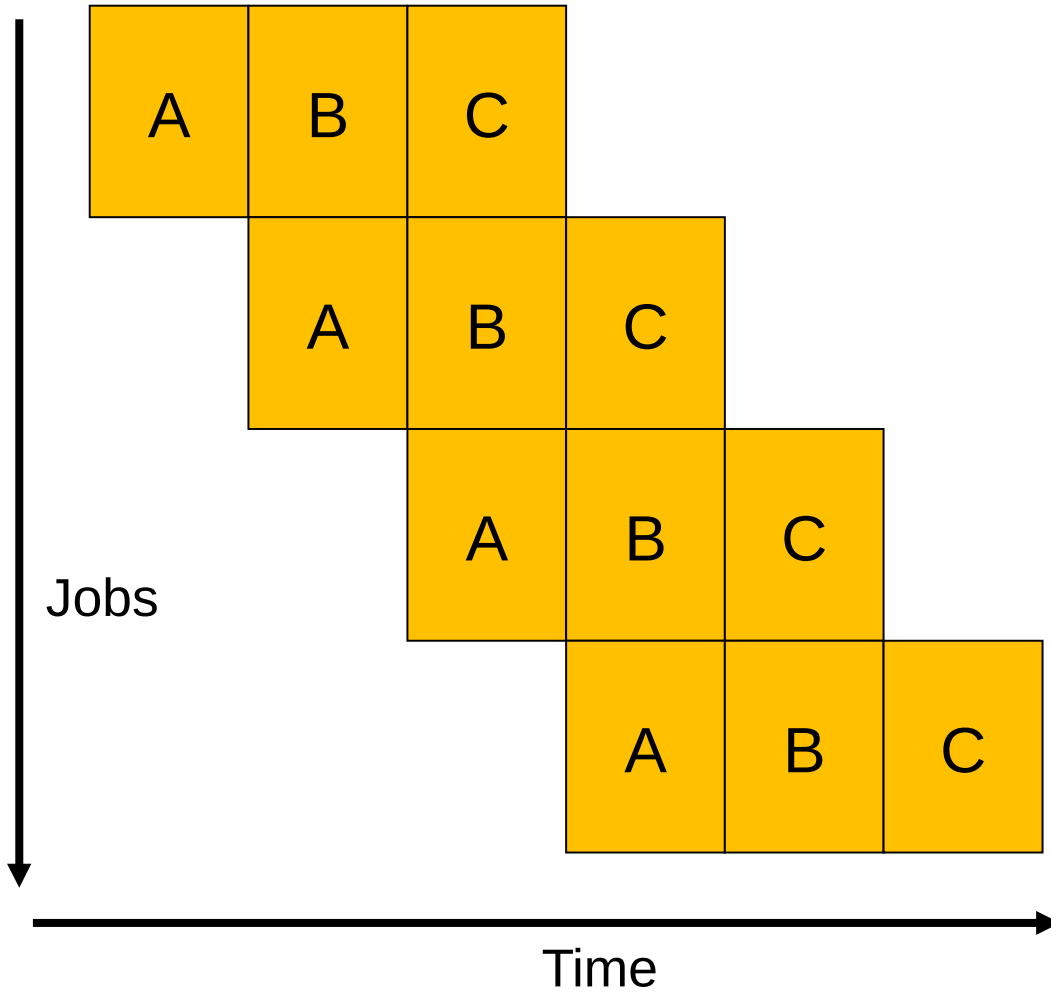
Start and finish a job before moving to the next



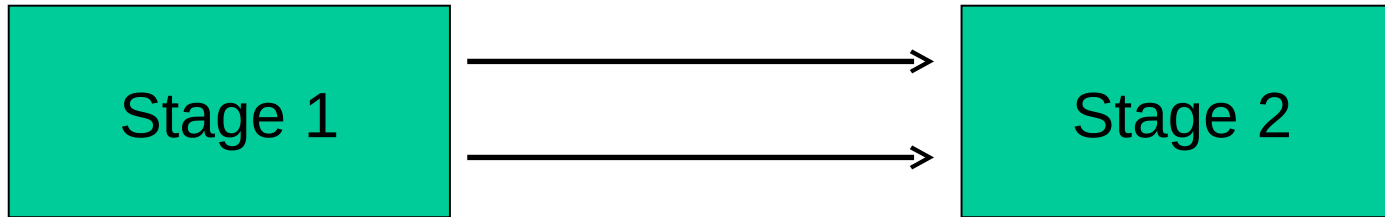
The Assembly Line

Pipelined

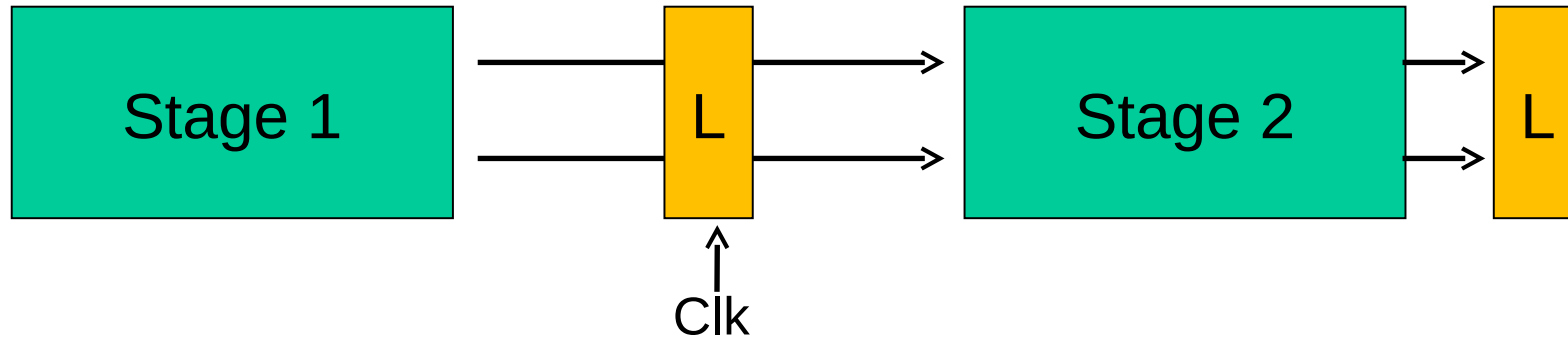
Break the job into smaller stages



Clocks and Latches



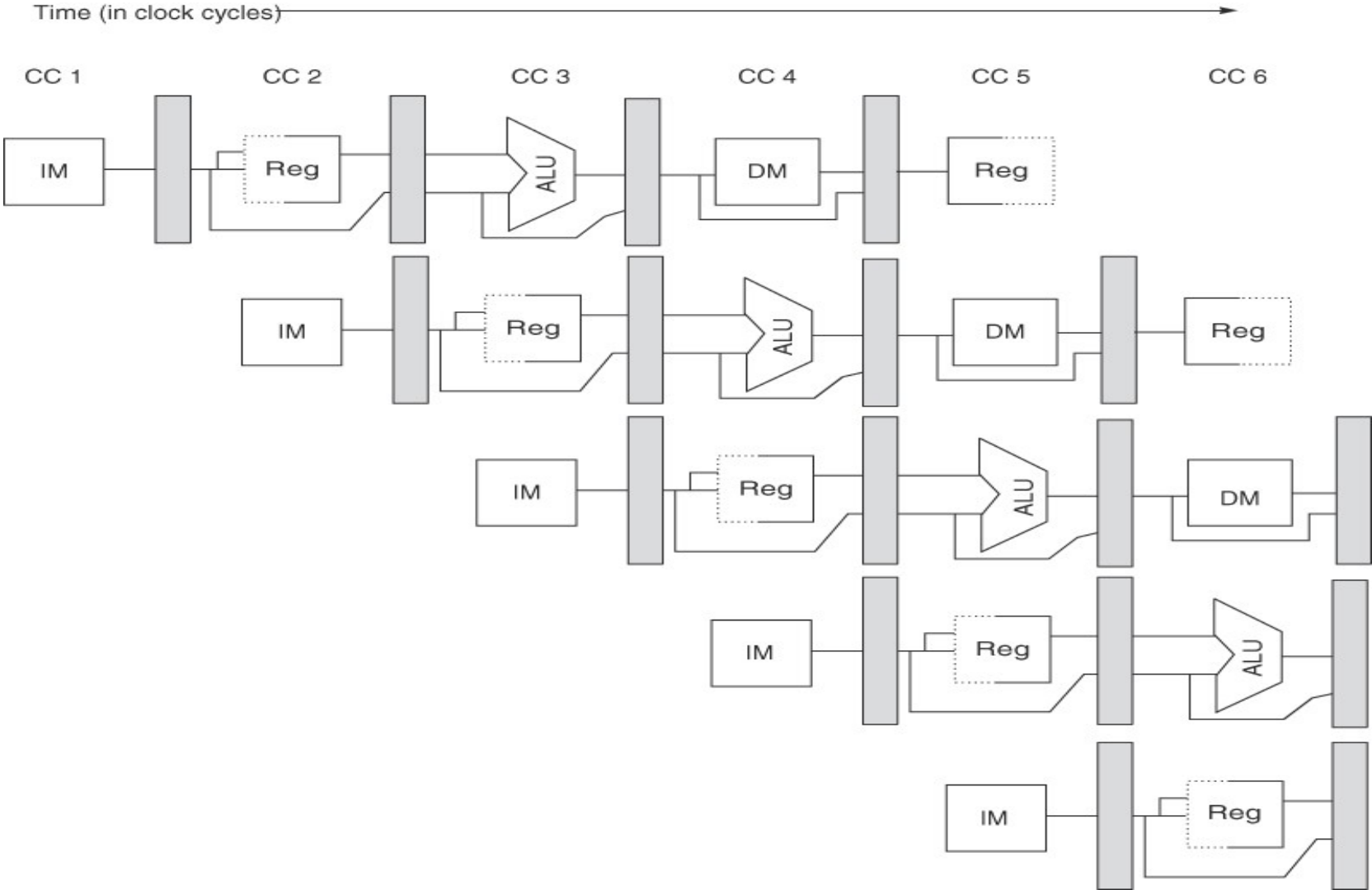
Clocks and Latches



Some Equations

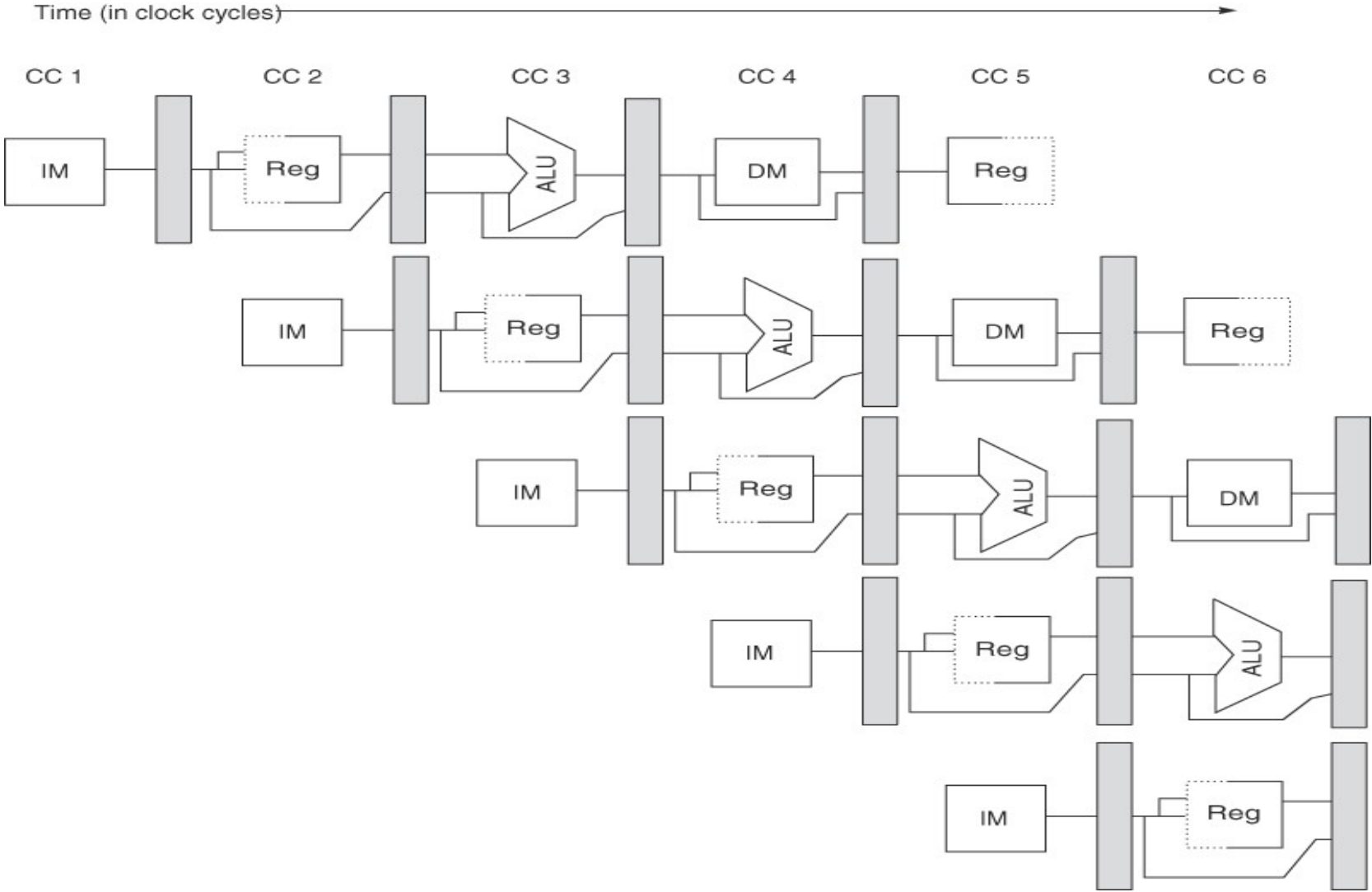
- Unpipelined: time to execute one instruction = $T + T_{ovh}$
- For an N-stage pipeline, time per stage = $T/N + T_{ovh}$
- Total time per instruction = $N (T/N + T_{ovh}) = T + N T_{ovh}$
- Clock cycle time = $T/N + T_{ovh}$
- Clock speed = $1 / (T/N + T_{ovh})$
- Ideal speedup = $(T + T_{ovh}) / (T/N + T_{ovh})$
- Cycles to complete one instruction = N
- Average CPI (cycles per instr) = 1

A 5-Stage Pipeline



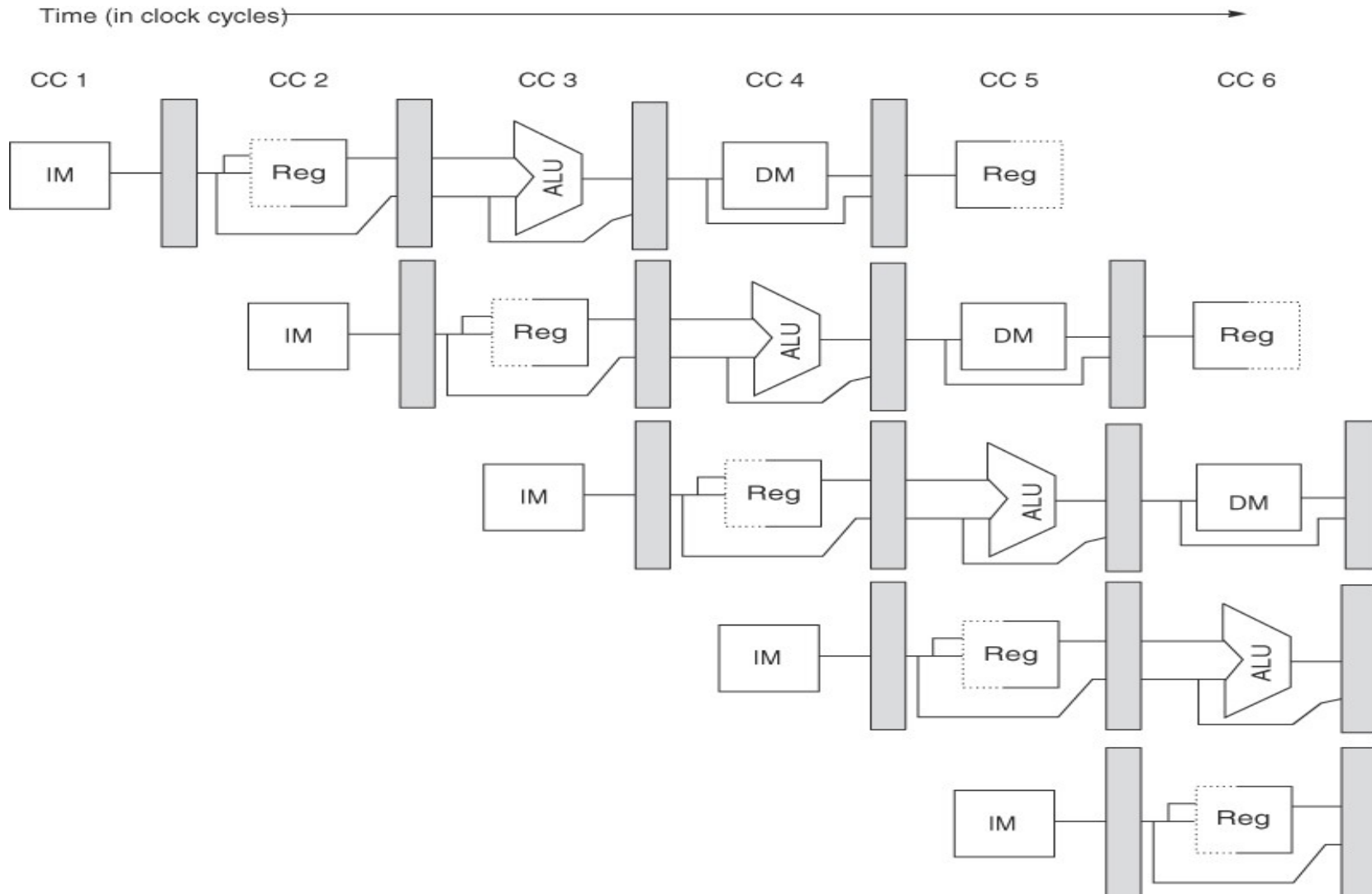
A 5-Stage Pipeline

Use the PC to access the I-cache and increment PC by 4



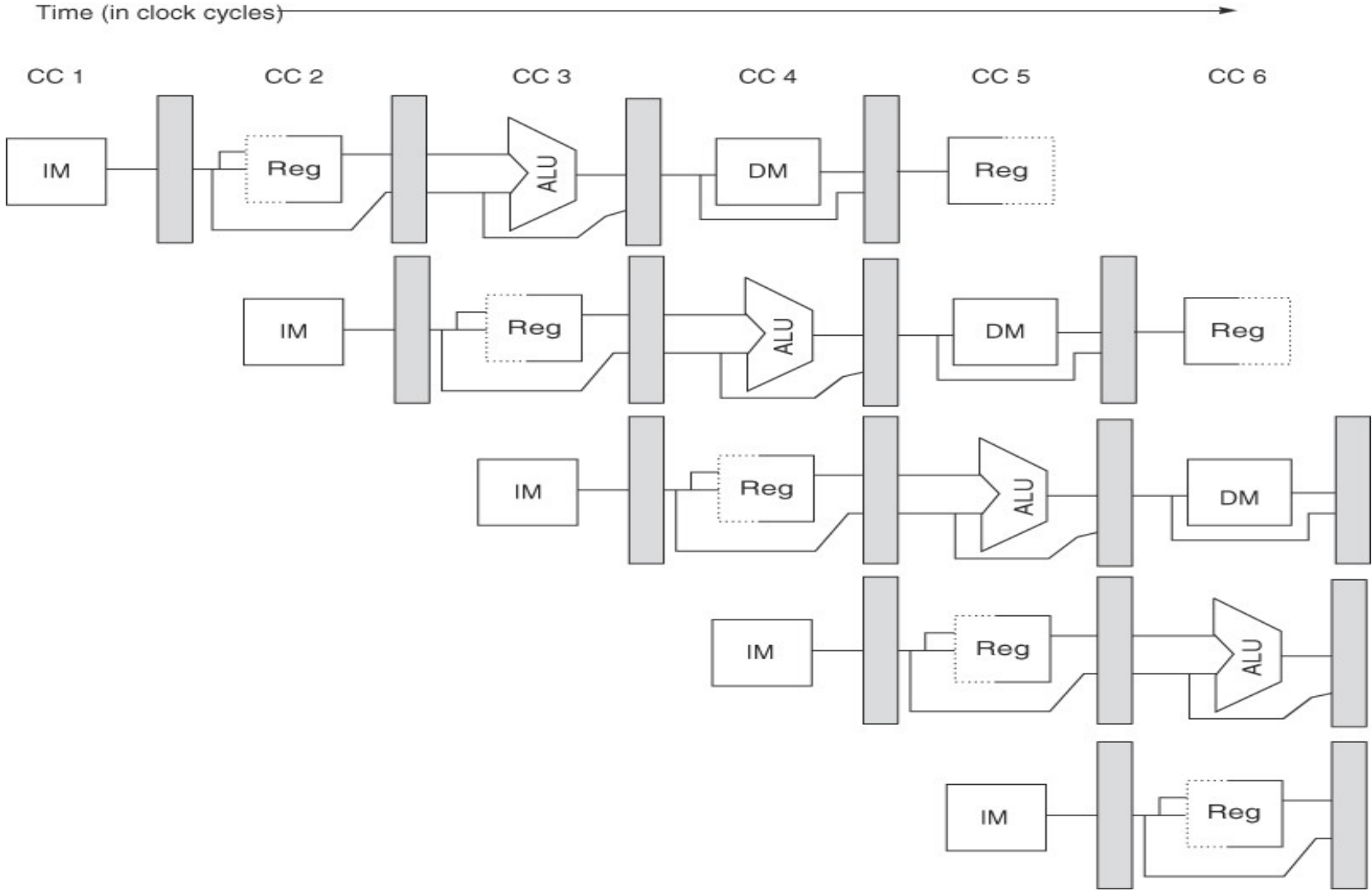
A 5-Stage Pipeline

Read registers, compare registers, compute branch target; for now, assume branches take 2 cyc (there is enough work that branches can easily take more)



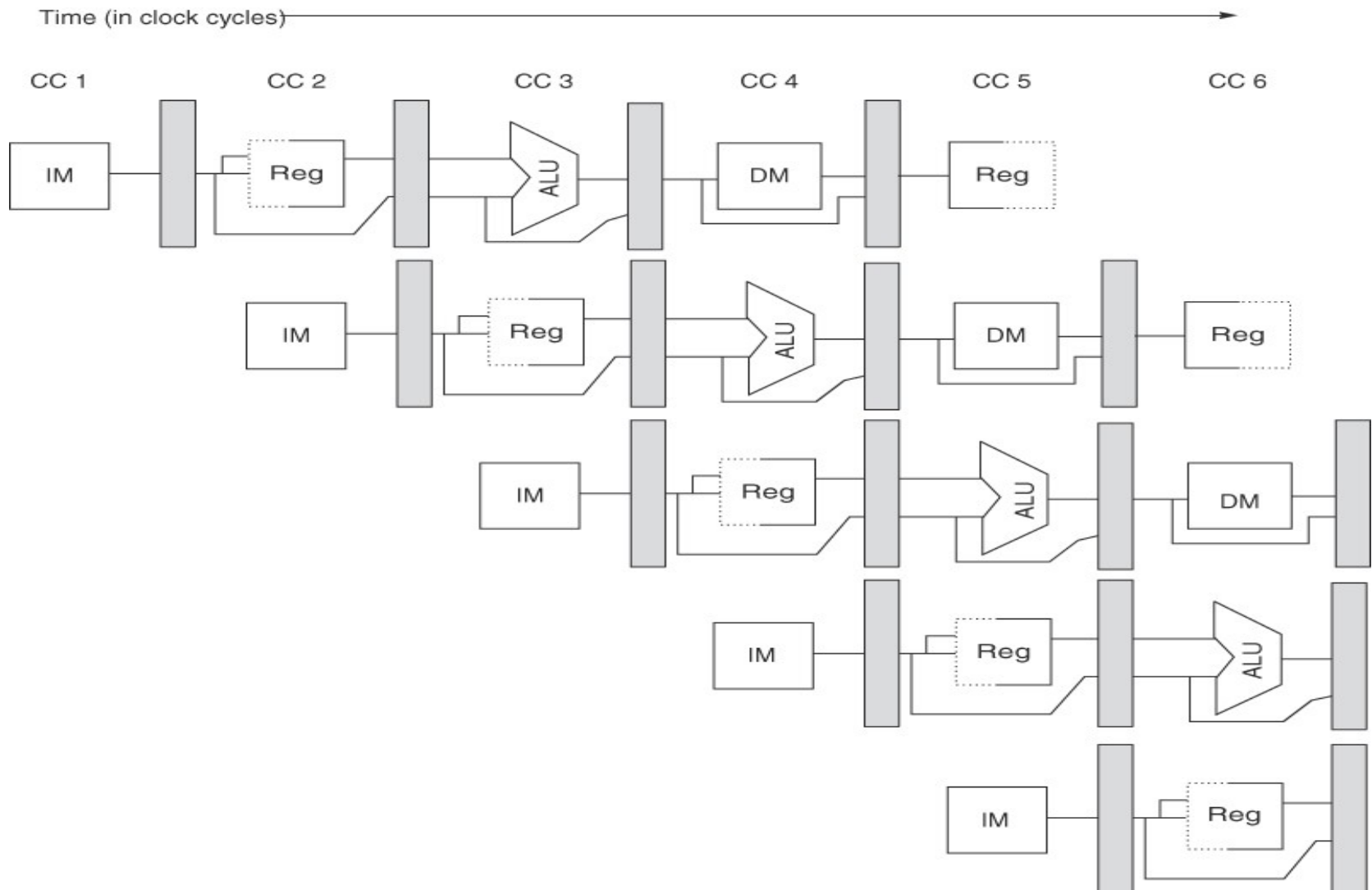
A 5-Stage Pipeline

ALU computation, effective address computation for load/store



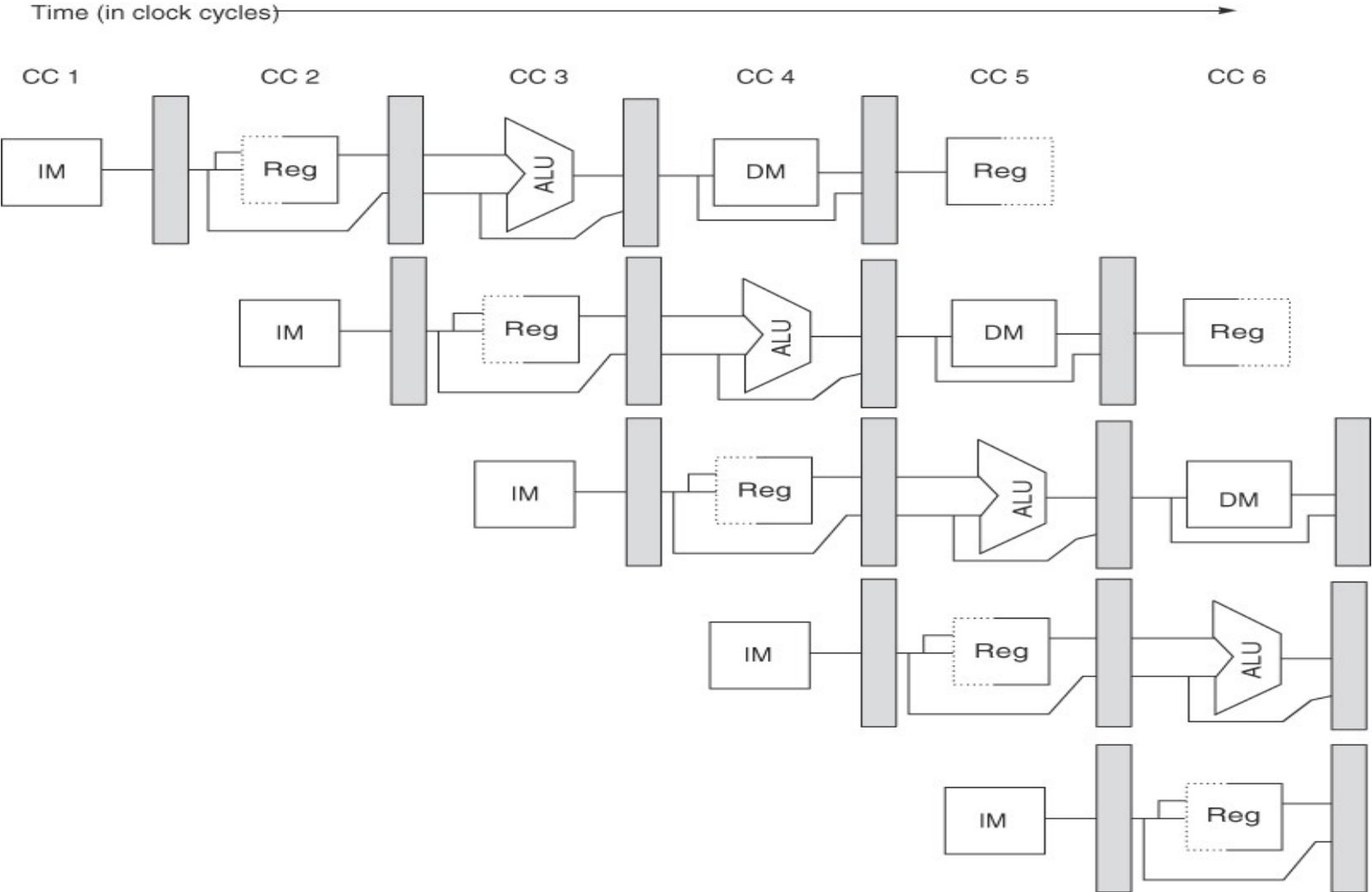
A 5-Stage Pipeline

Memory access to/from data cache, stores finish in 4 cycles



A 5-Stage Pipeline

Write result of ALU computation or load into register file



RISC/CISC Loads/Stores

Thank you!

AM vs. GM

- GM of IPCs = $1 / \text{GM of CPIs}$
- AM of IPCs represents thruput for a workload where each program runs sequentially for 1 cycle each; but high-IPC programs contribute more to the AM
- GM of IPCs does not represent run-time for any real workload (what does it mean to multiply instructions?); but every program's IPC contributes equally to the final measure

Speedup Vs. Percentage

- “Speedup” is a ratio = old exec time / new exec time
- “Improvement”, “Increase”, “Decrease” usually refer to percentage relative to the baseline
= (new perf – old perf) / old perf
- A program ran in 100 seconds on my old laptop and in 70 seconds on my new laptop
 - What is the speedup?
 - What is the percentage increase in performance?
 - What is the reduction in execution time?