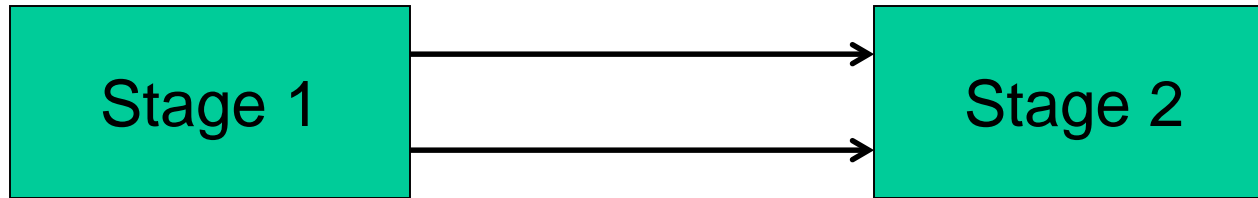


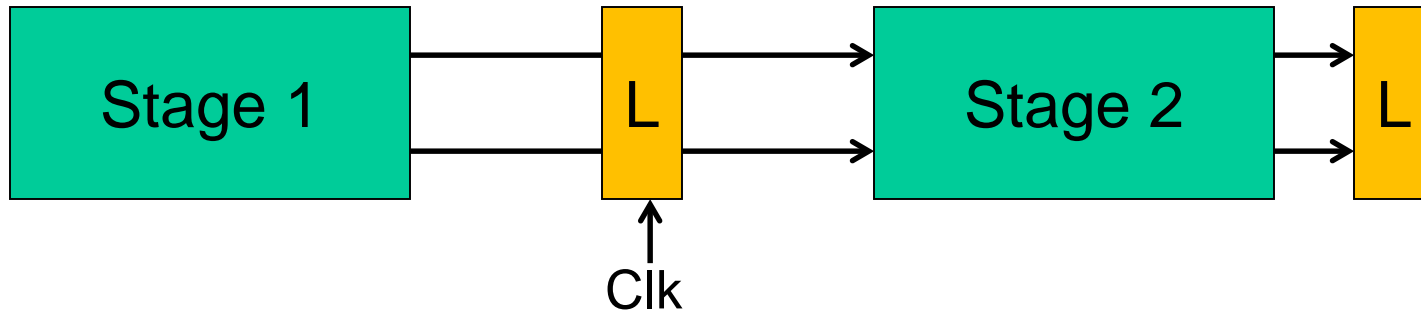
Lecture: Pipelining Basics

- Topics: Basic pipelining implementation

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

Problem 1

- An unpipelined processor takes 5 ns to work on one instruction. It then takes 0.2 ns to latch its results into latches. I was able to convert the circuits into 5 equal sequential pipeline stages. Answer the following, assuming that there are no stalls in the pipeline.
 - What are the cycle times in the two processors?
 - What are the clock speeds?
 - What are the IPCs?
 - How long does it take to finish one instr?
 - What is the speedup from pipelining?

Problem 1

- An unpipelined processor takes 5 ns to work on one instruction. It then takes 0.2 ns to latch its results into latches. I was able to convert the circuits into 5 equal sequential pipeline stages. Answer the following, assuming that there are no stalls in the pipeline.
- What are the cycle times in the two processors?
5.2ns and 1.2ns
- What are the clock speeds? 192 MHz and 833 MHz
- What are the IPCs? 1 and 1
- How long does it take to finish one instr? 5.2ns and 6ns
- What is the speedup from pipelining? $833/192 = 4.34$

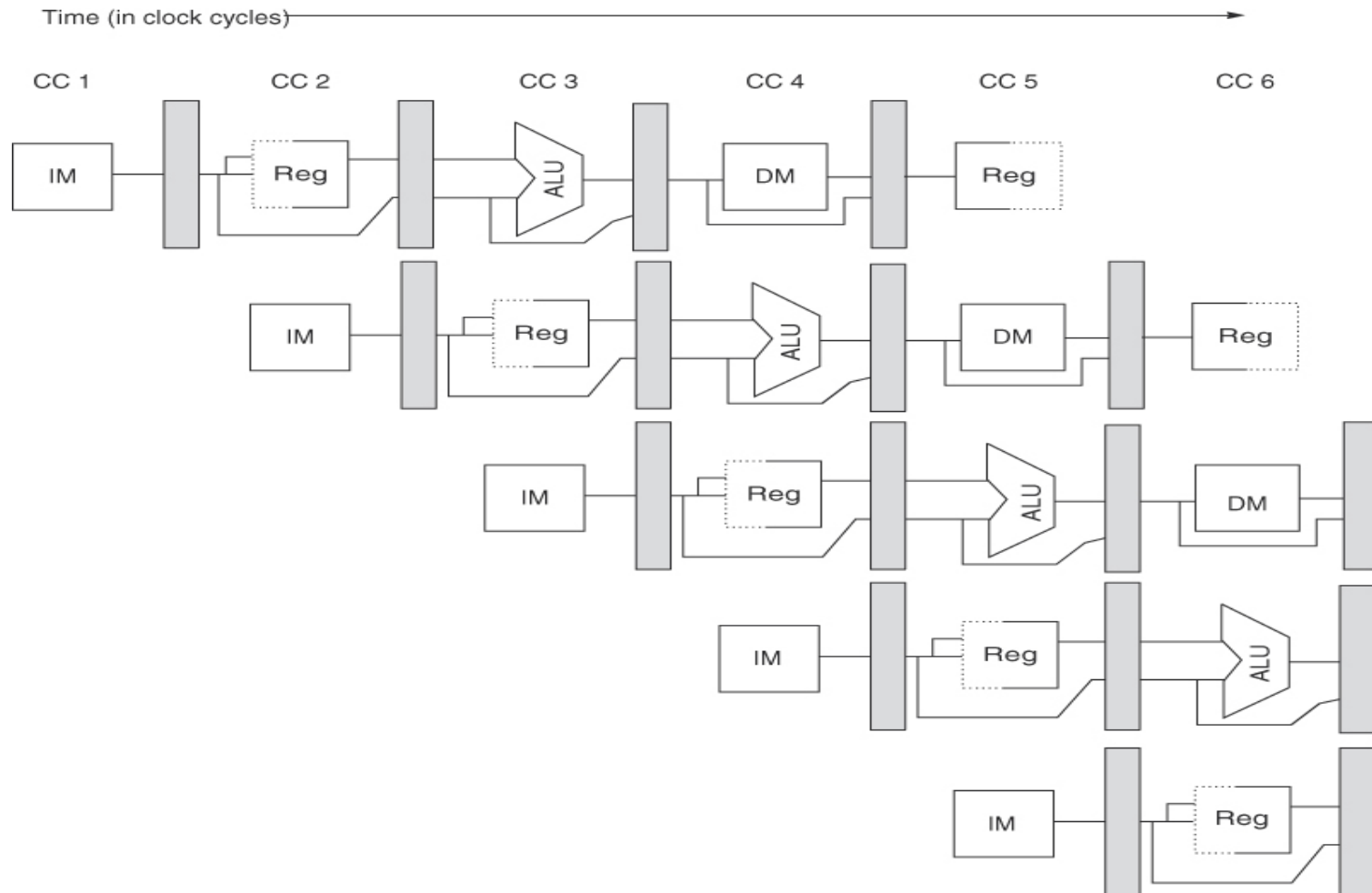
Problem 2

- An unpipelined processor takes 5 ns to work on one instruction. It then takes 0.2 ns to latch its results into latches. I was able to convert the circuits into 5 sequential pipeline stages. The stages have the following lengths: 1ns; 0.6ns; 1.2ns; 1.4ns; 0.8ns. Answer the following, assuming that there are no stalls in the pipeline.
 - What is the cycle time in the new processor?
 - What is the clock speed?
 - What is the IPC?
 - How long does it take to finish one instr?
 - What is the speedup from pipelining?
 - What is the max speedup from pipelining?

Problem 2

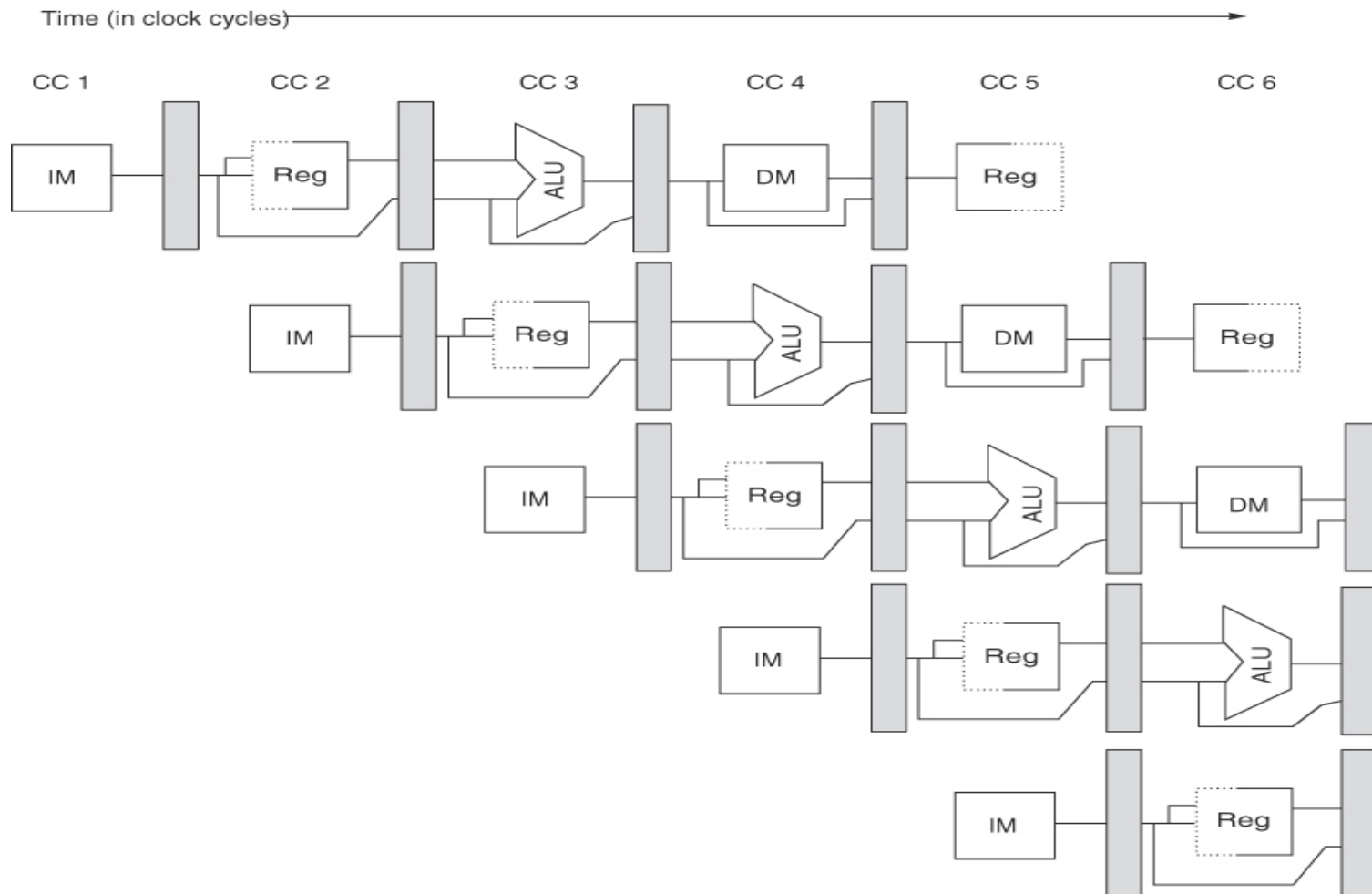
- An unpipelined processor takes 5 ns to work on one instruction. It then takes 0.2 ns to latch its results into latches. I was able to convert the circuits into 5 sequential pipeline stages. The stages have the following lengths: 1ns; 0.6ns; 1.2ns; 1.4ns; 0.8ns. Answer the following, assuming that there are no stalls in the pipeline.
 - What is the cycle time in the new processor? 1.6ns
 - What is the clock speed? 625 MHz
 - What is the IPC? 1
 - How long does it take to finish one instr? 8ns
 - What is the speedup from pipelining? $625/192 = 3.26$
 - What is the max speedup from pipelining? $5.2/0.2 = 26$

A 5-Stage Pipeline



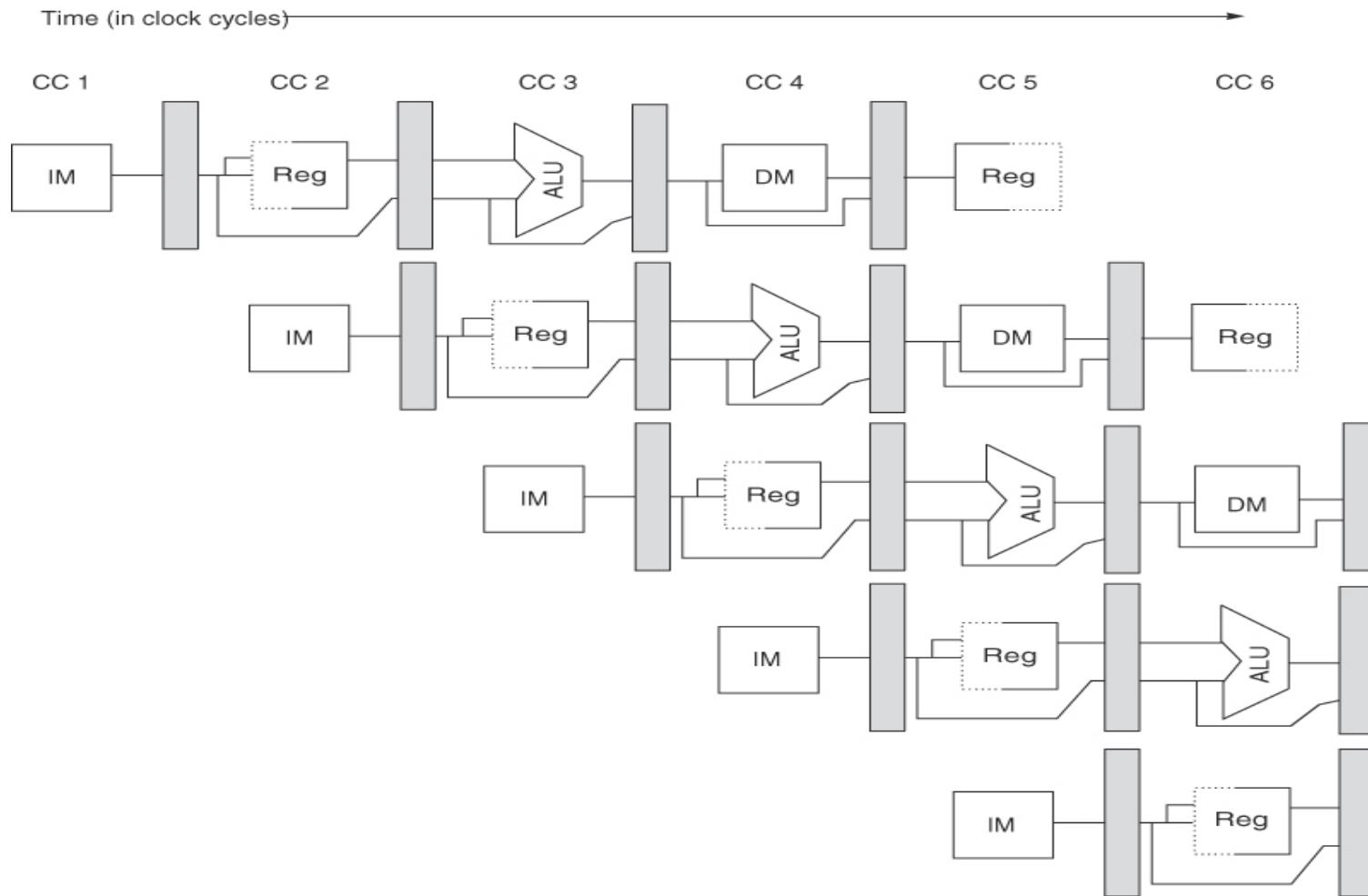
A 5-Stage Pipeline

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



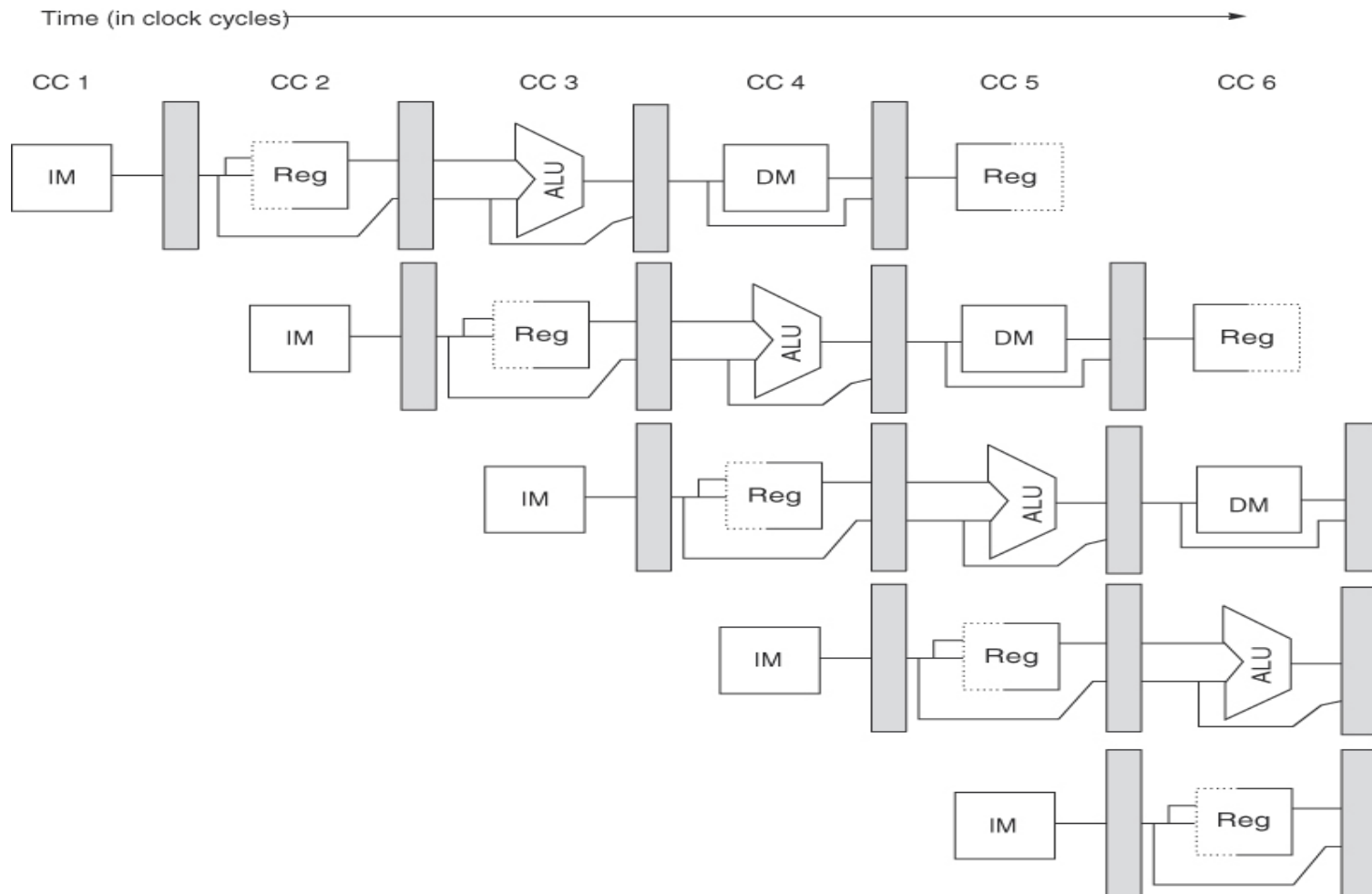
A 5-Stage Pipeline

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



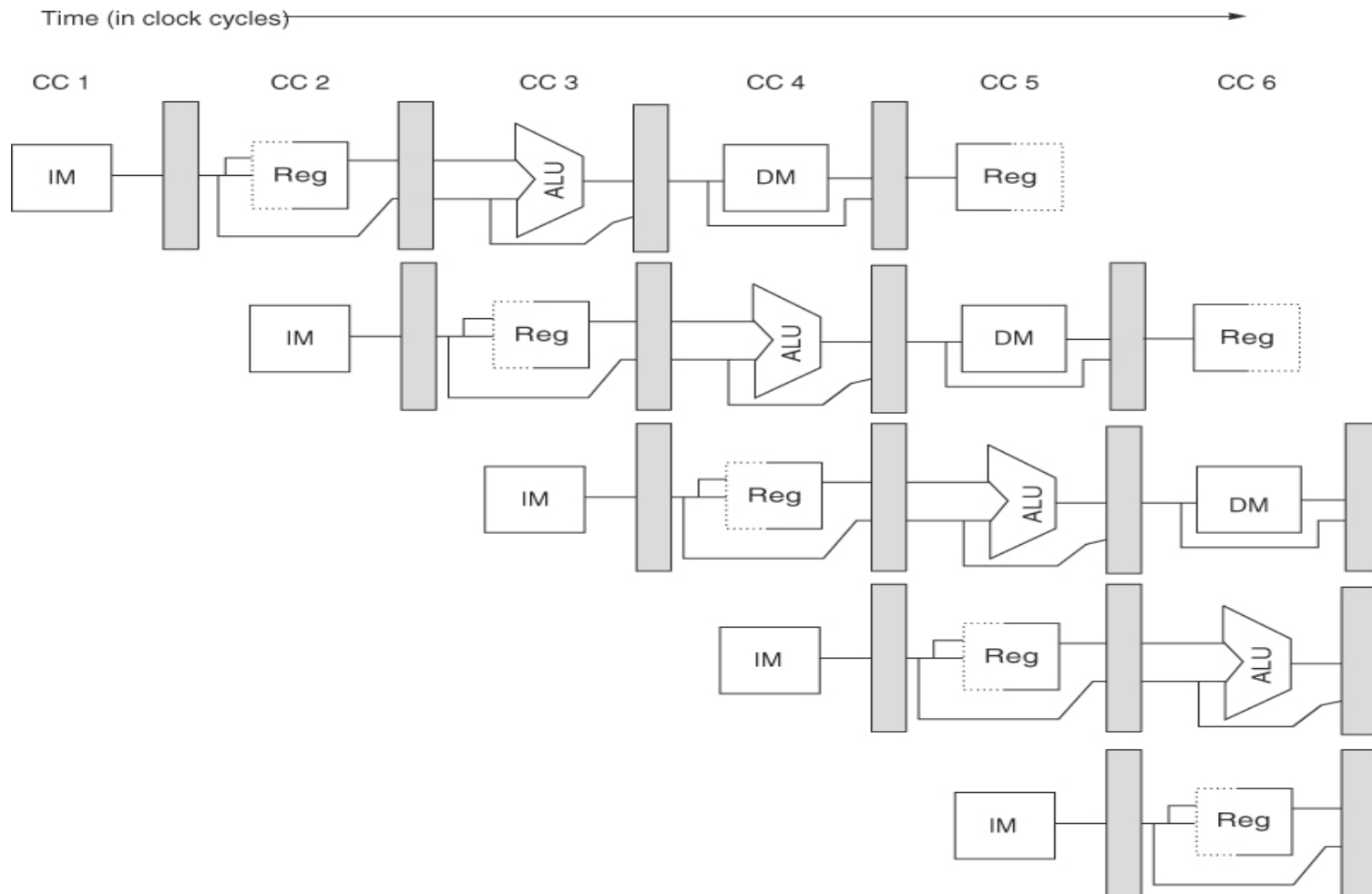
A 5-Stage Pipeline

ALU computation, effective address computation for load/store



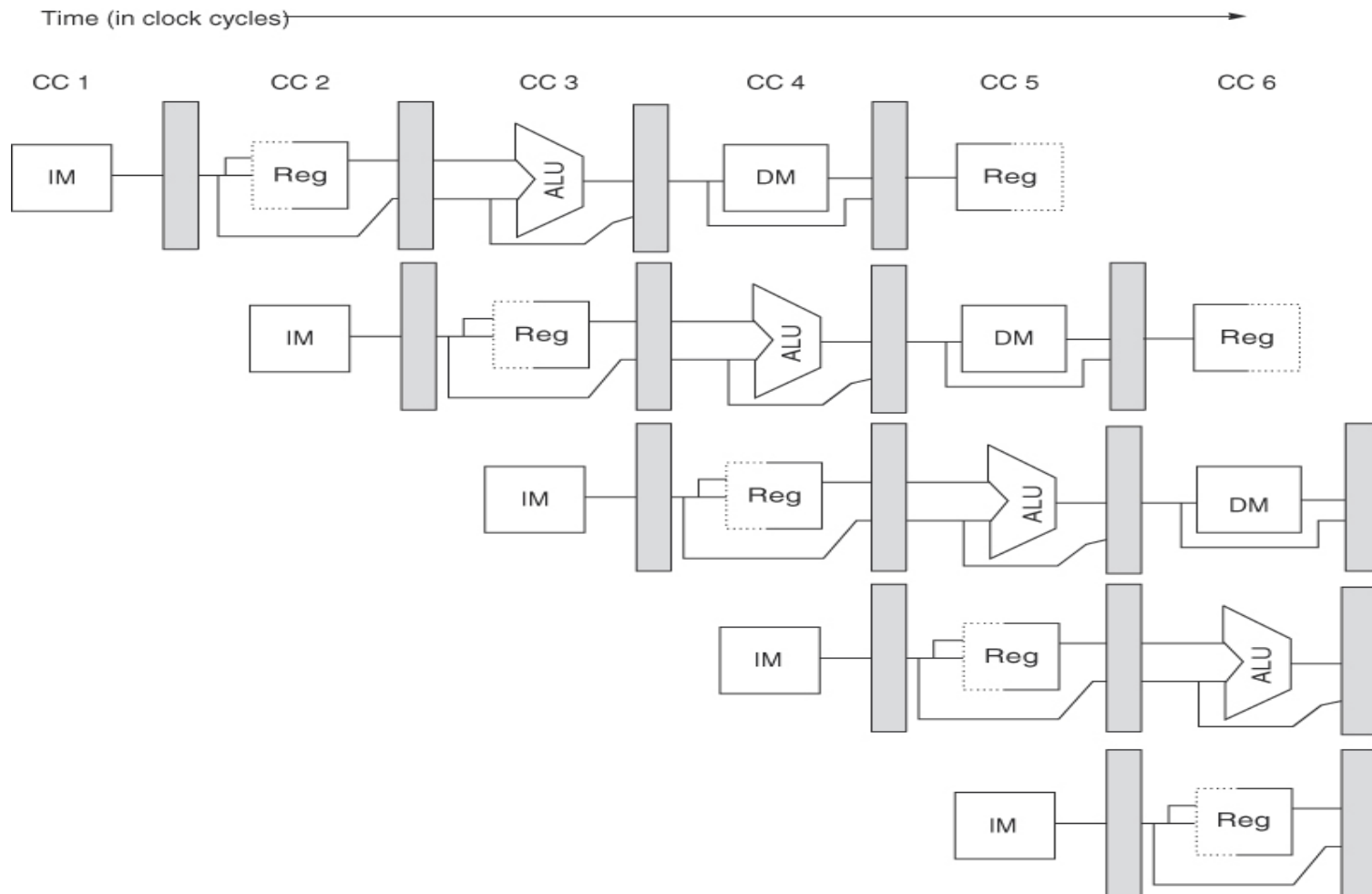
A 5-Stage Pipeline

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

Registers and memory

Complex and reduced instrs 15

Format of a load/store

Title

- Bullet