

Computer Architecture

Pipeline Hazards

Hazard

- instructions interfere with each other

Hazard

- instructions interfere with each other
- a.k.a. *conflicts*

3 Types of Hazards

1. data
2. control
3. structural

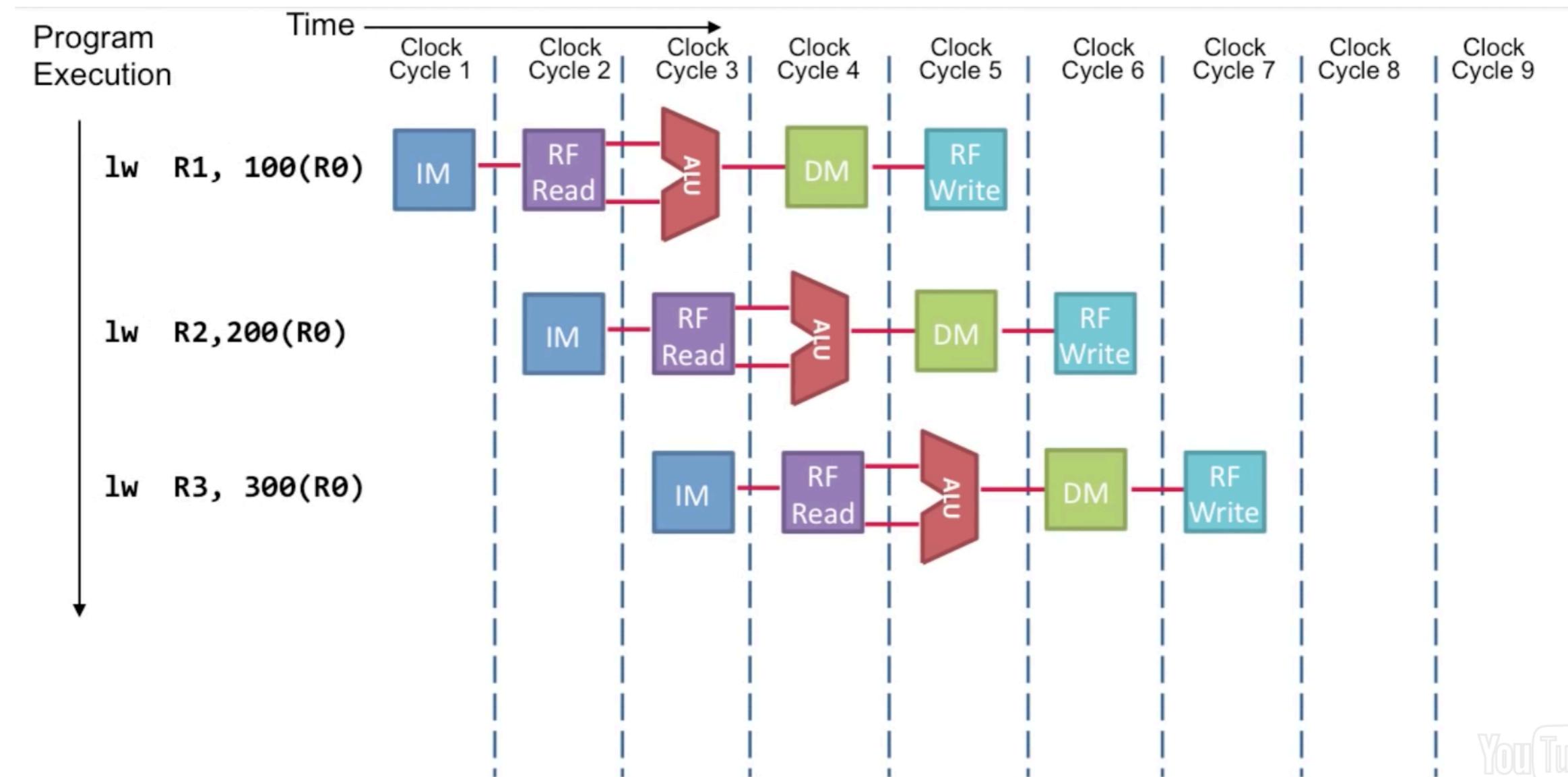
3 Types of Hazards

1. data - different instructions need result (or data) from other instructions
2. control - different instructions execute depending on others
3. structural - different instructions need the same resources

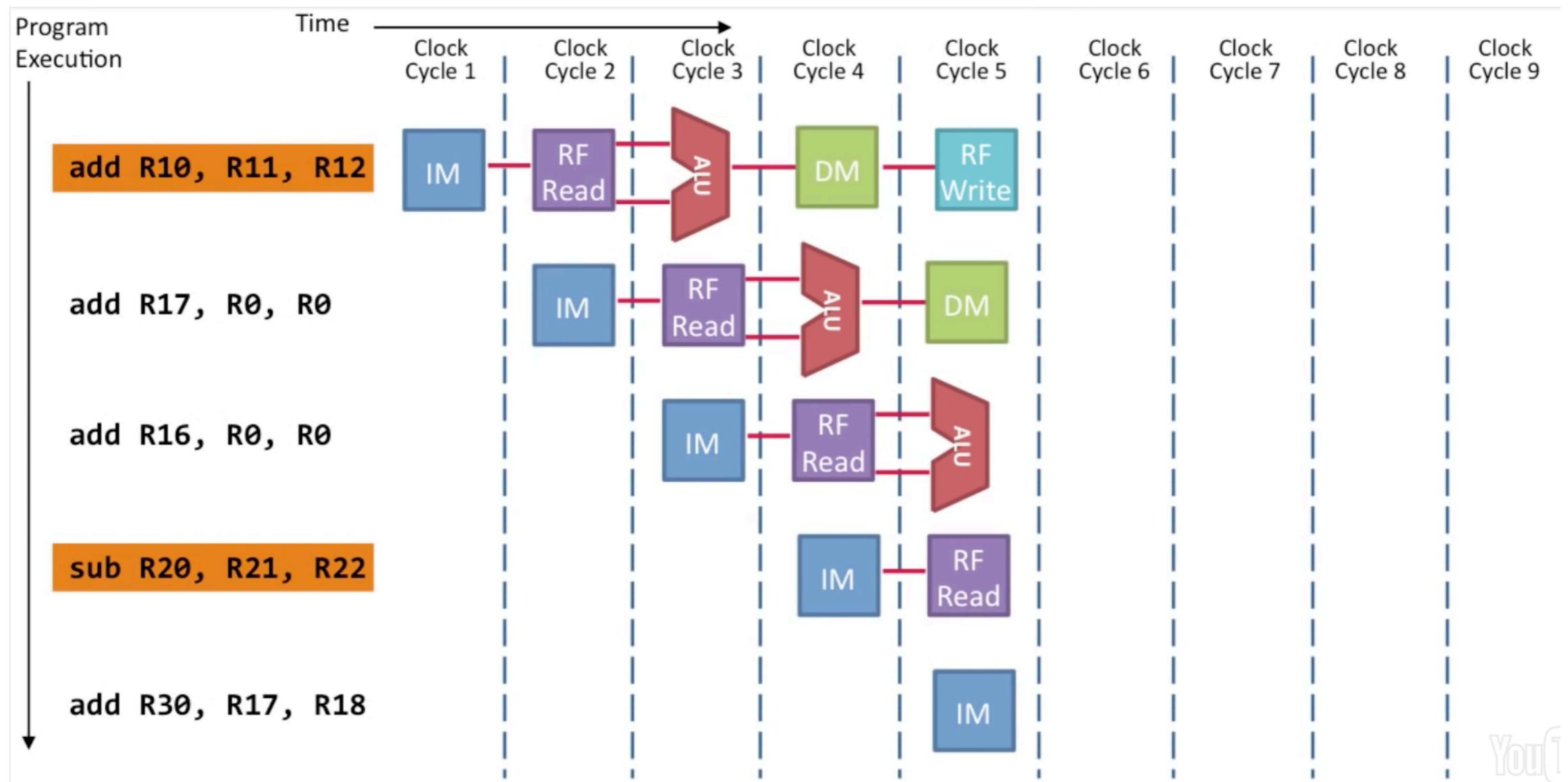
3 Types of Hazards

1. data - different instructions need result (or data) from other instructions
2. control - different instructions execute depending on others
3. structural - different instructions need the same resources

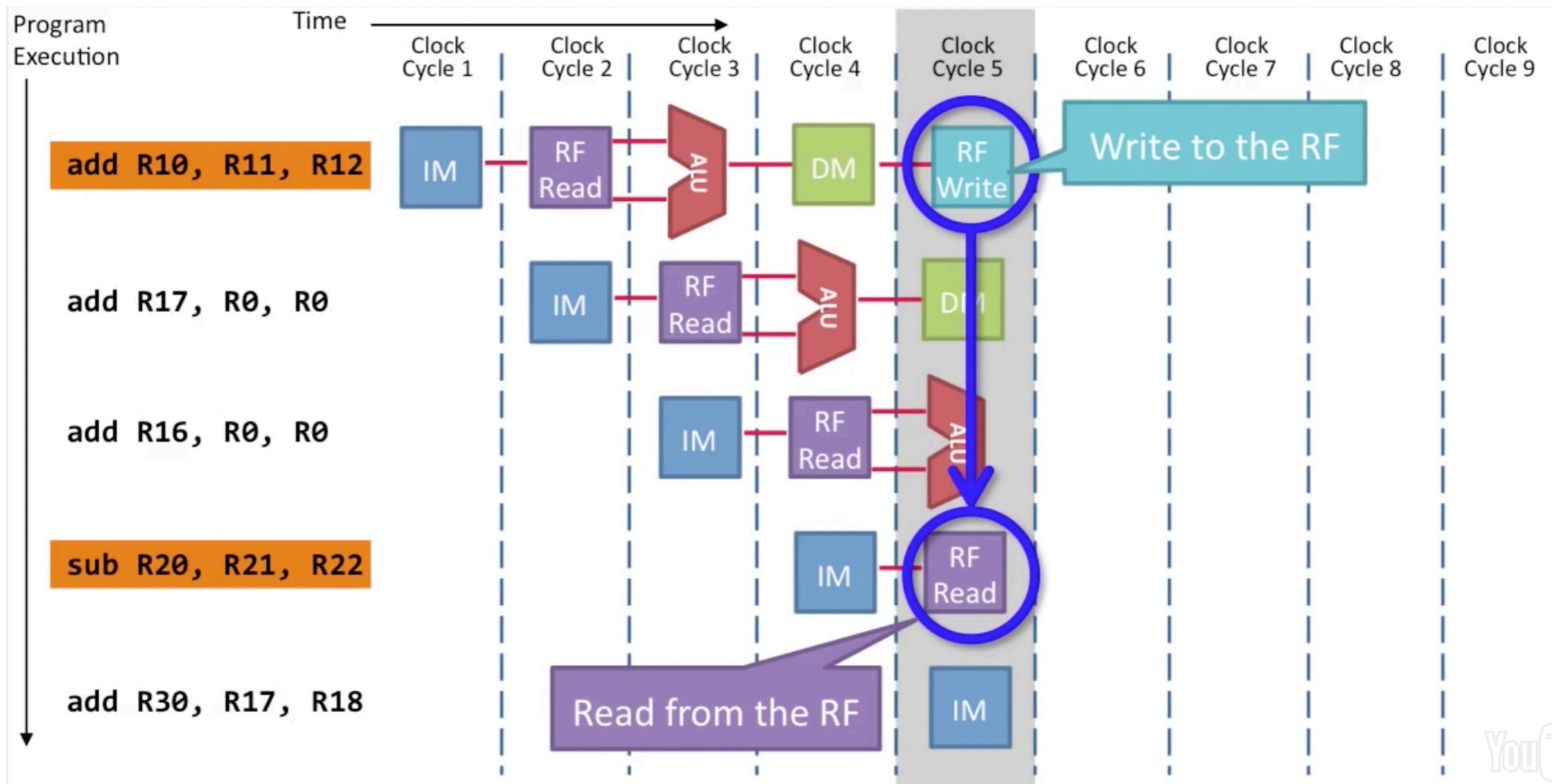
Good pipelining...



Interference...

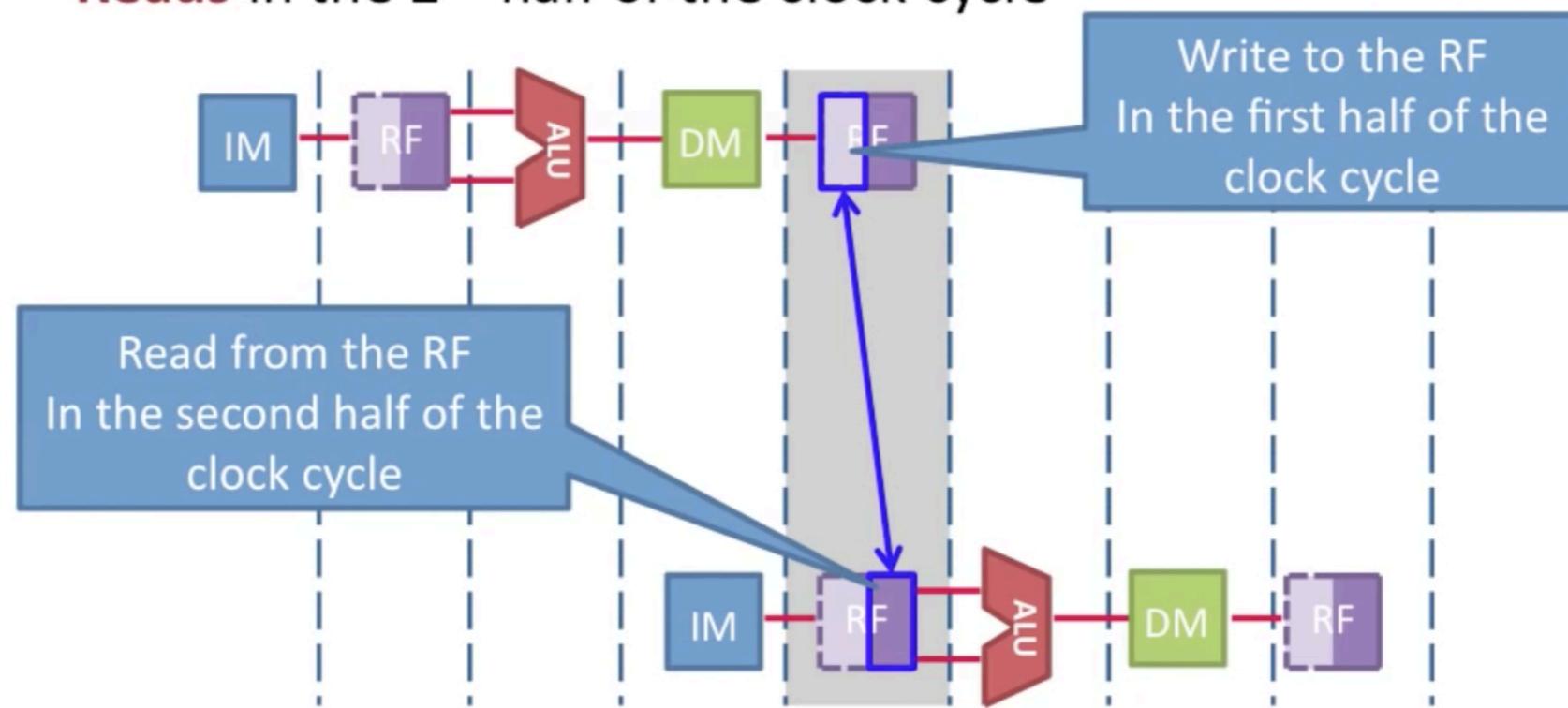


Interference...

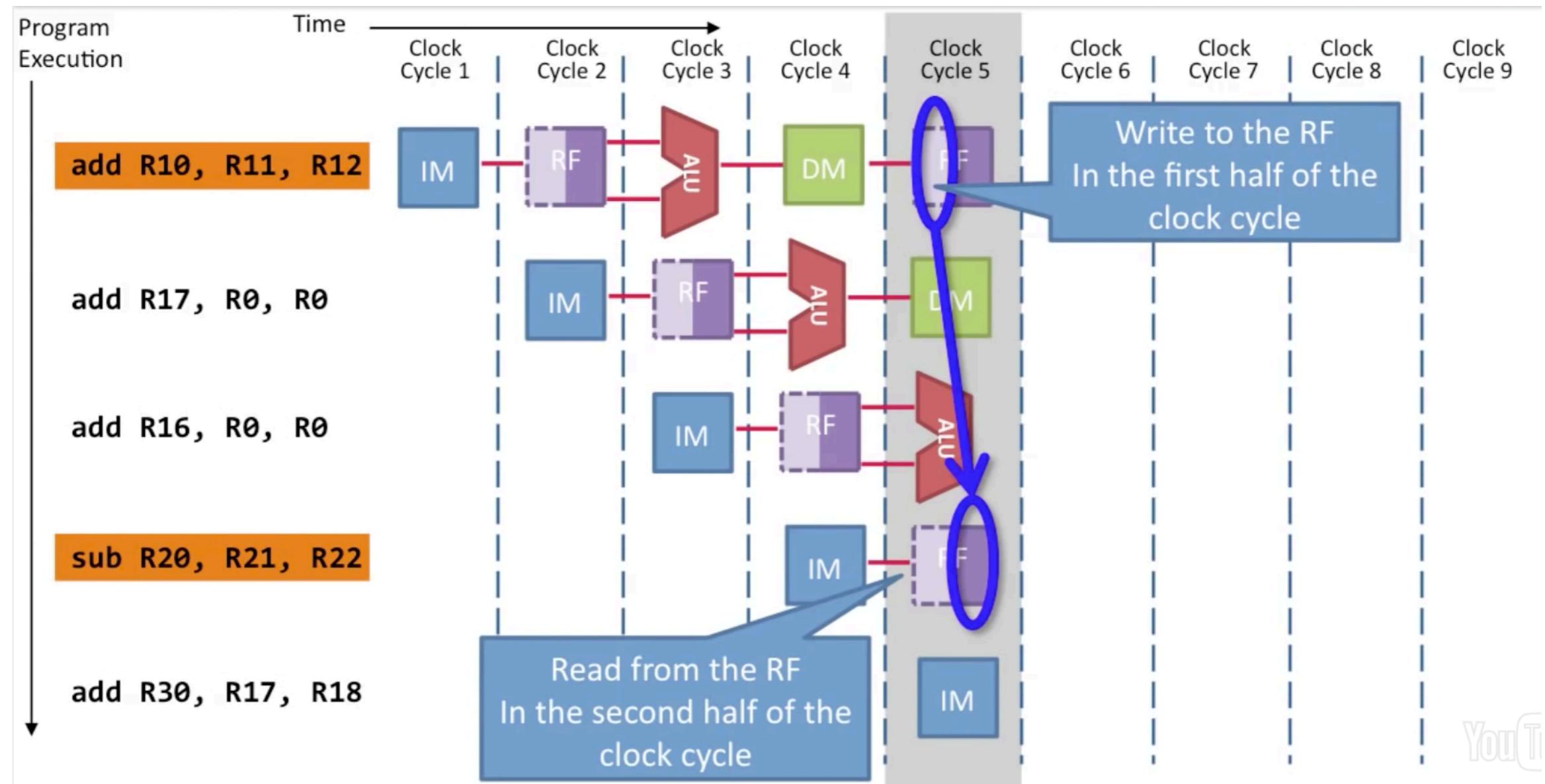


Solution is double-pumped register file...

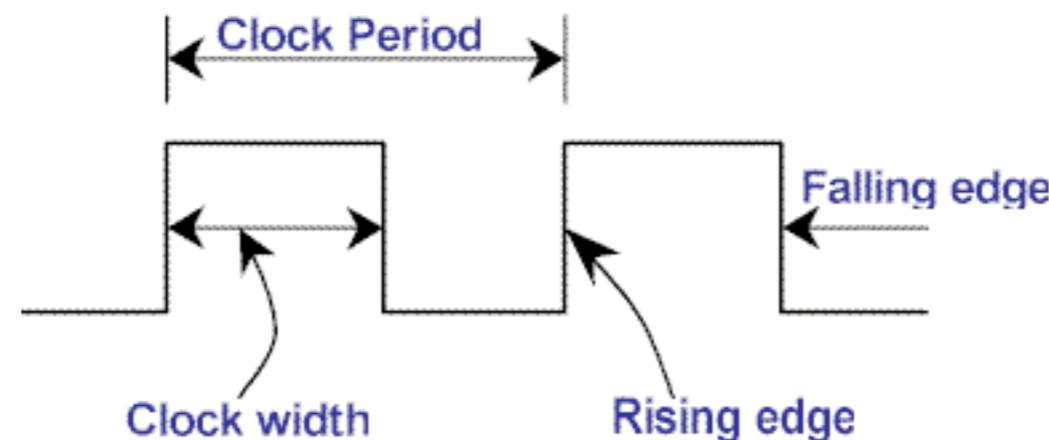
- **Writes** in the 1st half of the clock cycle
- **Reads** in the 2nd half of the clock cycle



Solution is double-pumped register file...

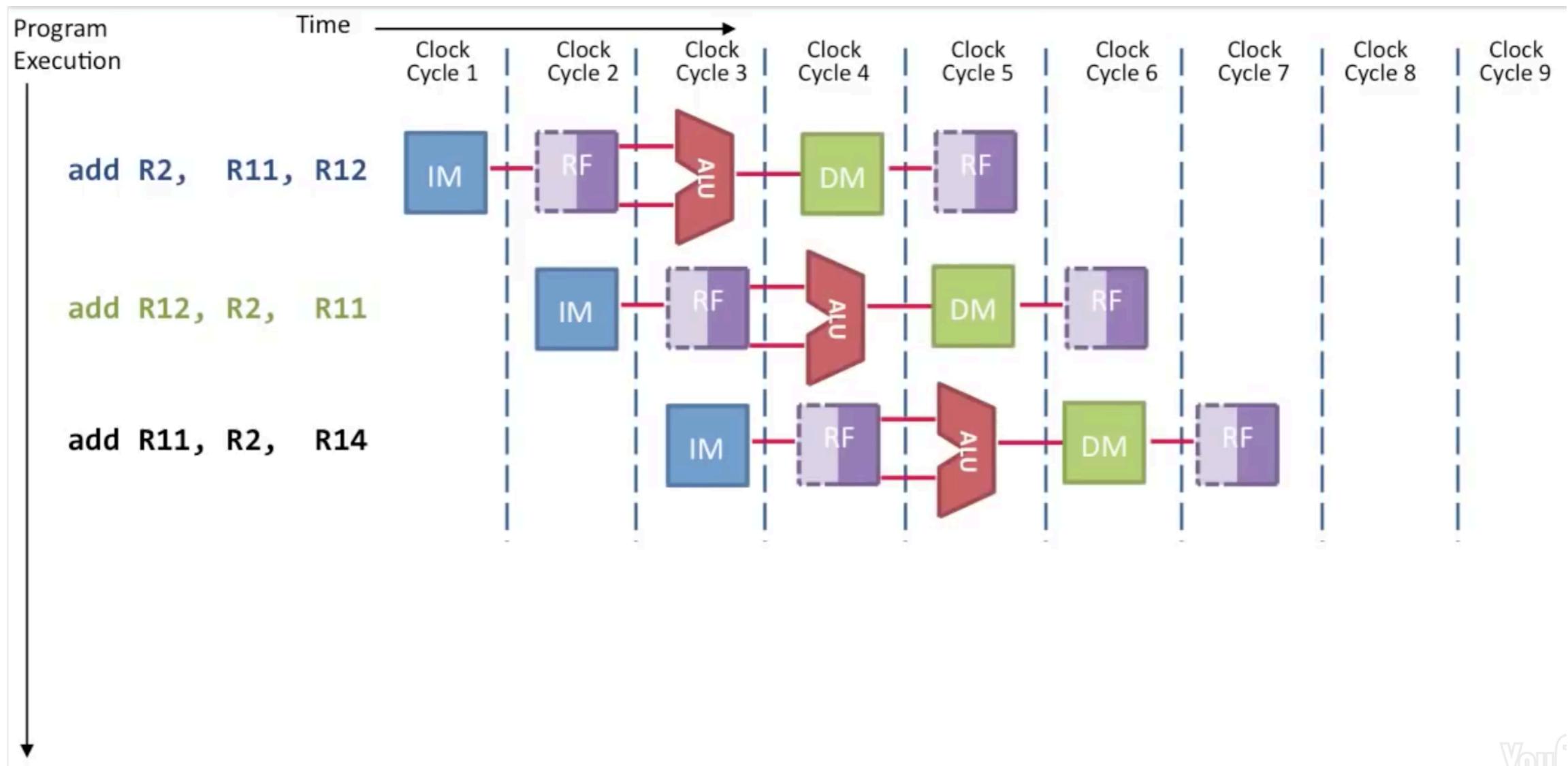


The following figure shows the clock cycle or clock period for a MIPS processor,

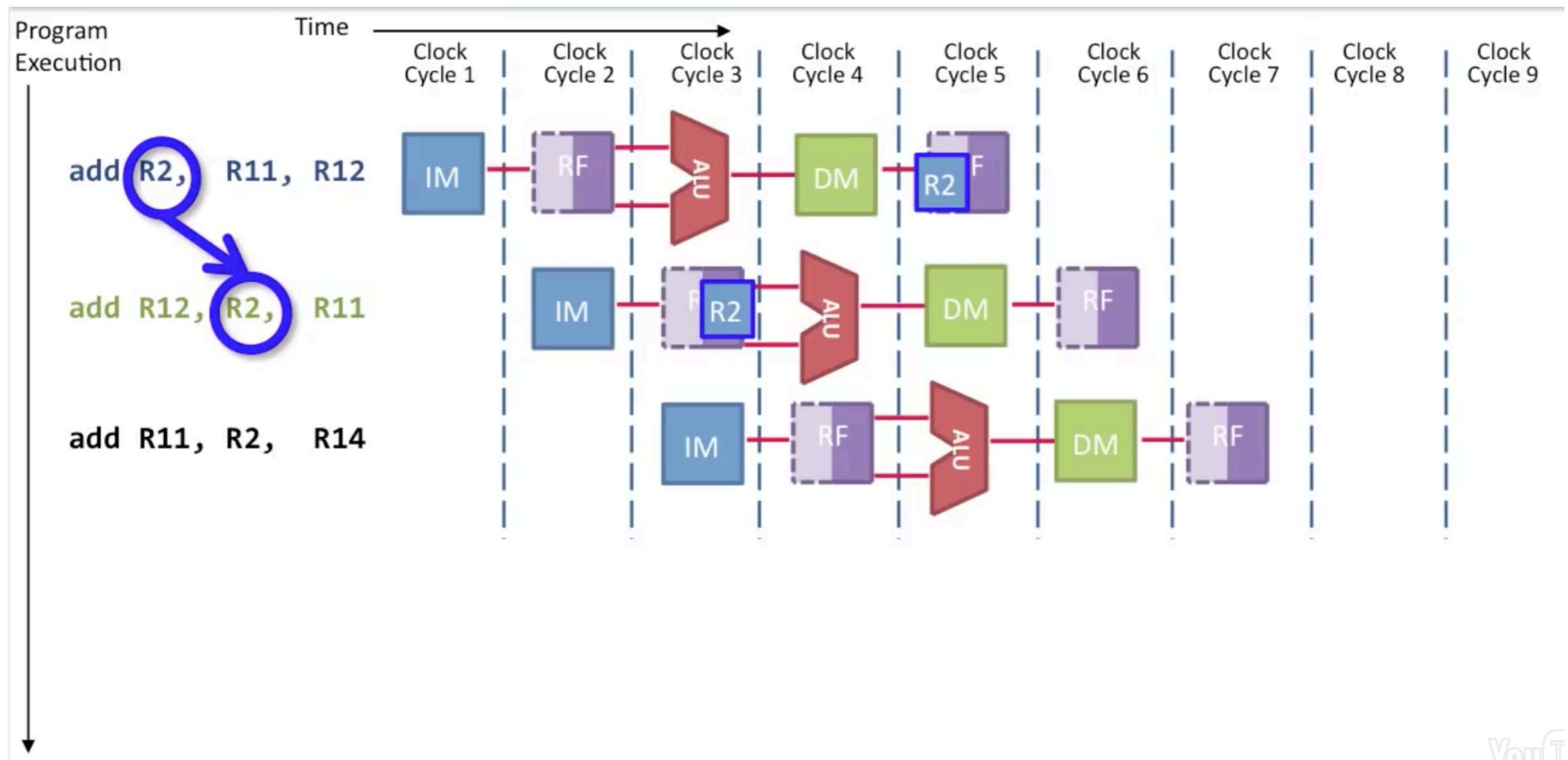


Question: What is the signal value of the clock when the double-pumped register file performs data read? 0 or 1?

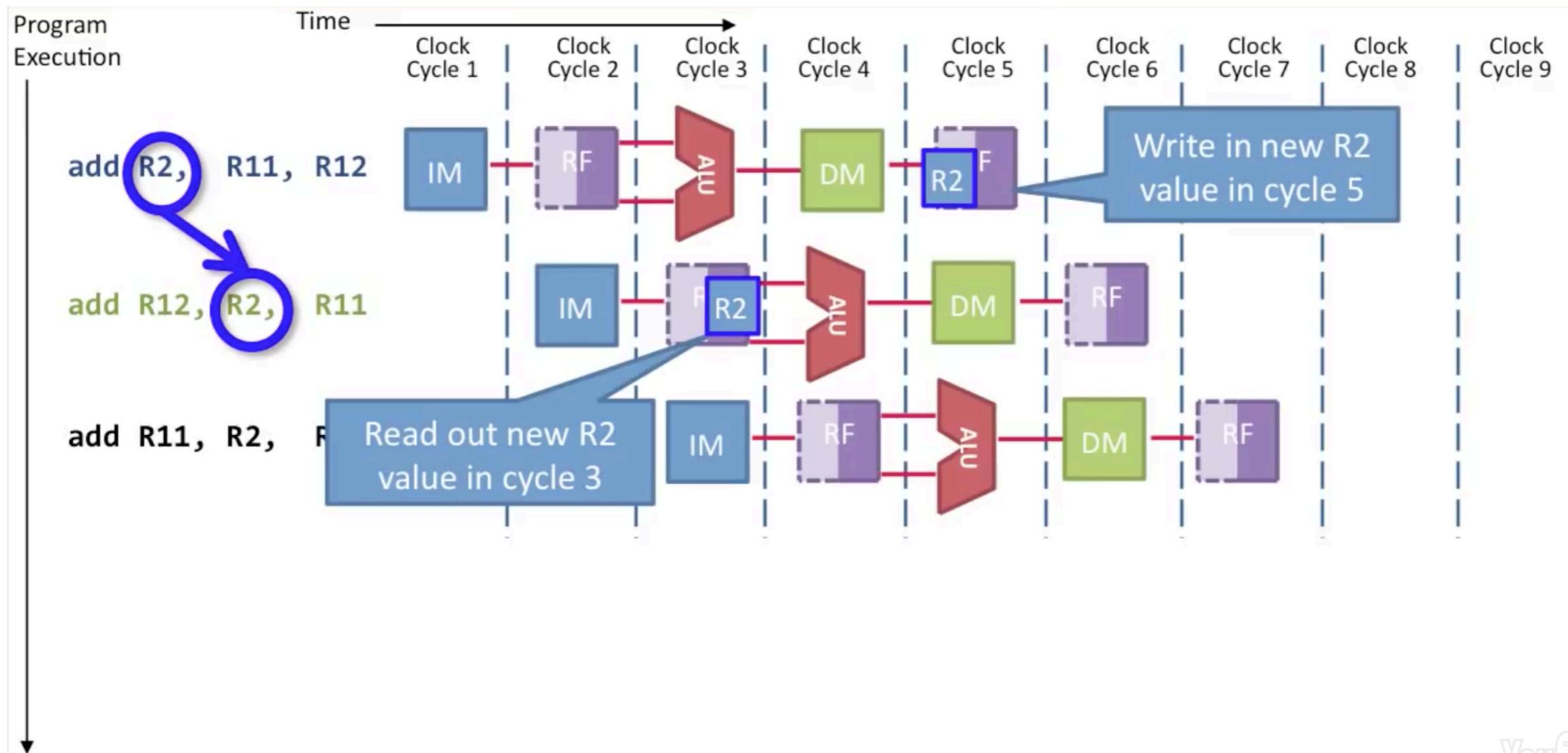
Data Hazards



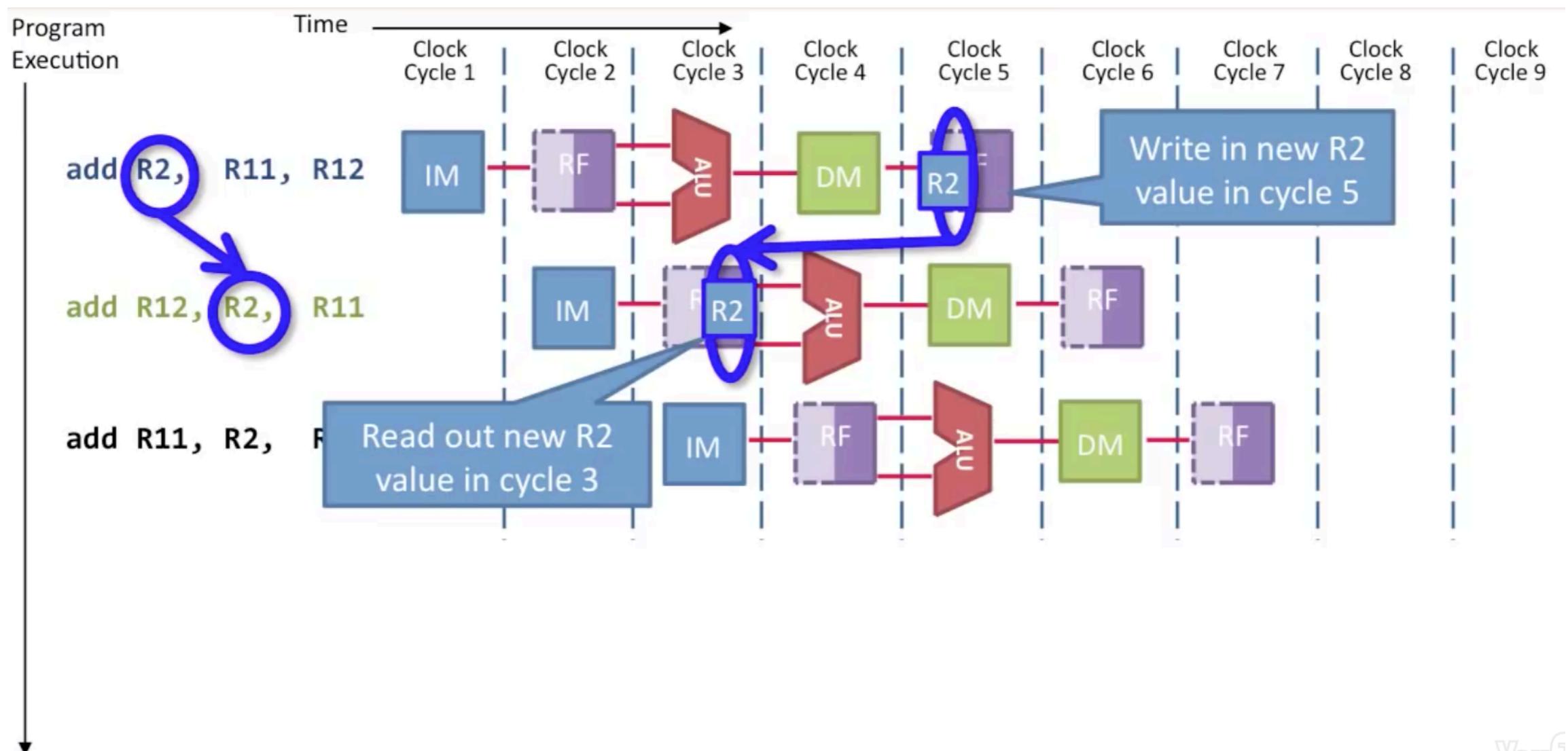
Data Hazards



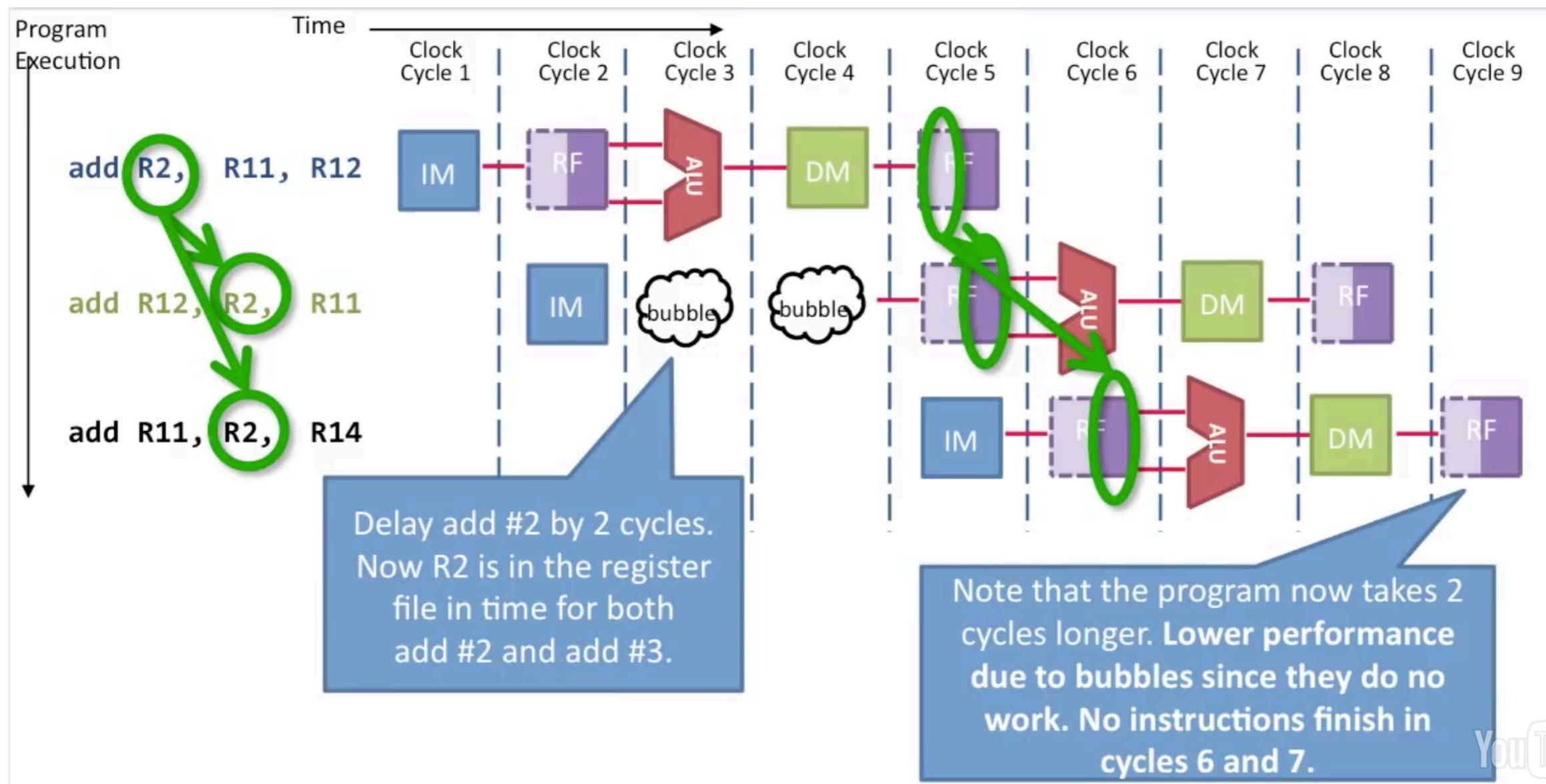
Data Hazards



Data Hazards



Data Hazards



Consider the following MIPS instructions and count the number of data hazards present.

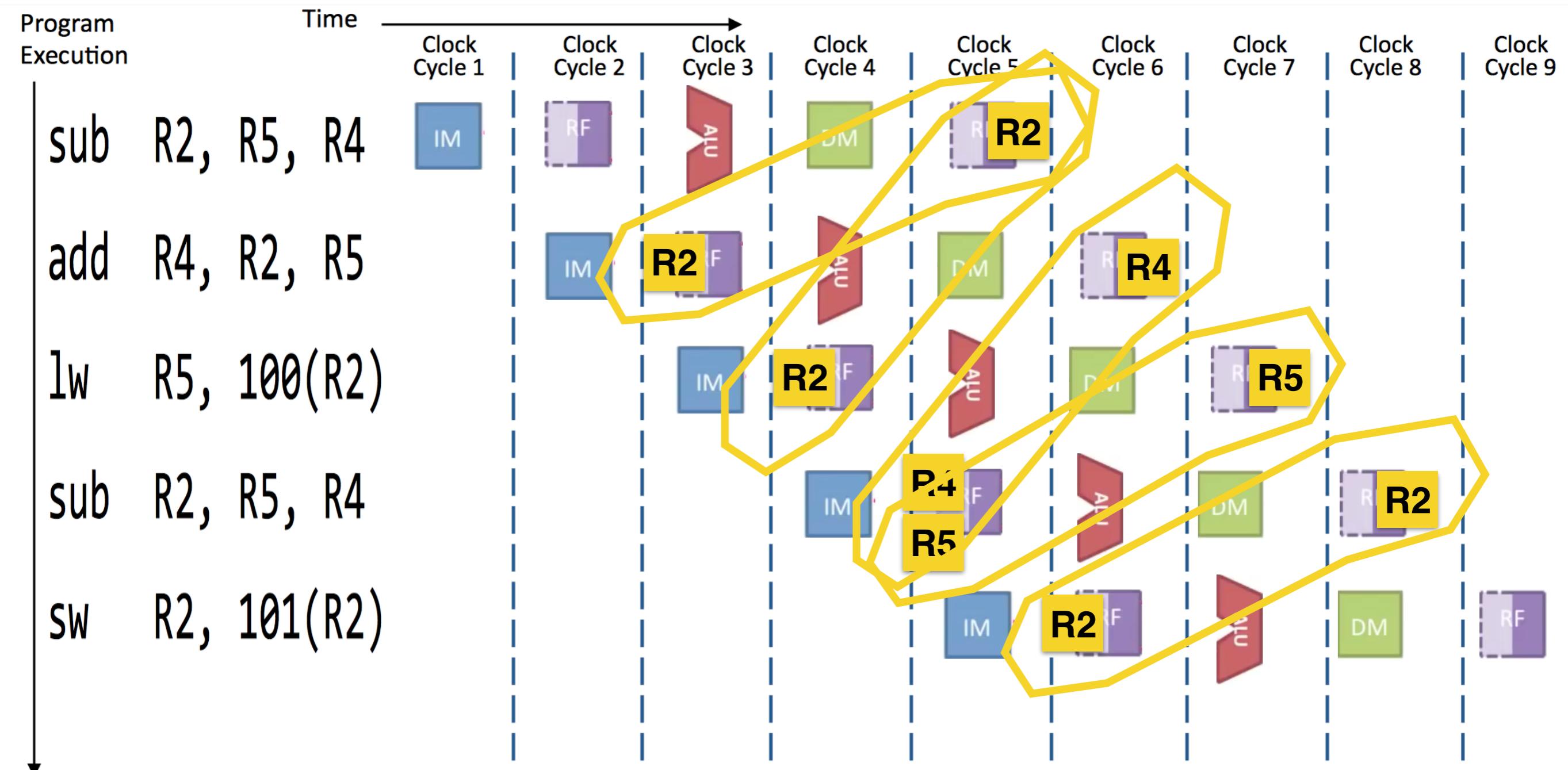
sub R2, R5, R4

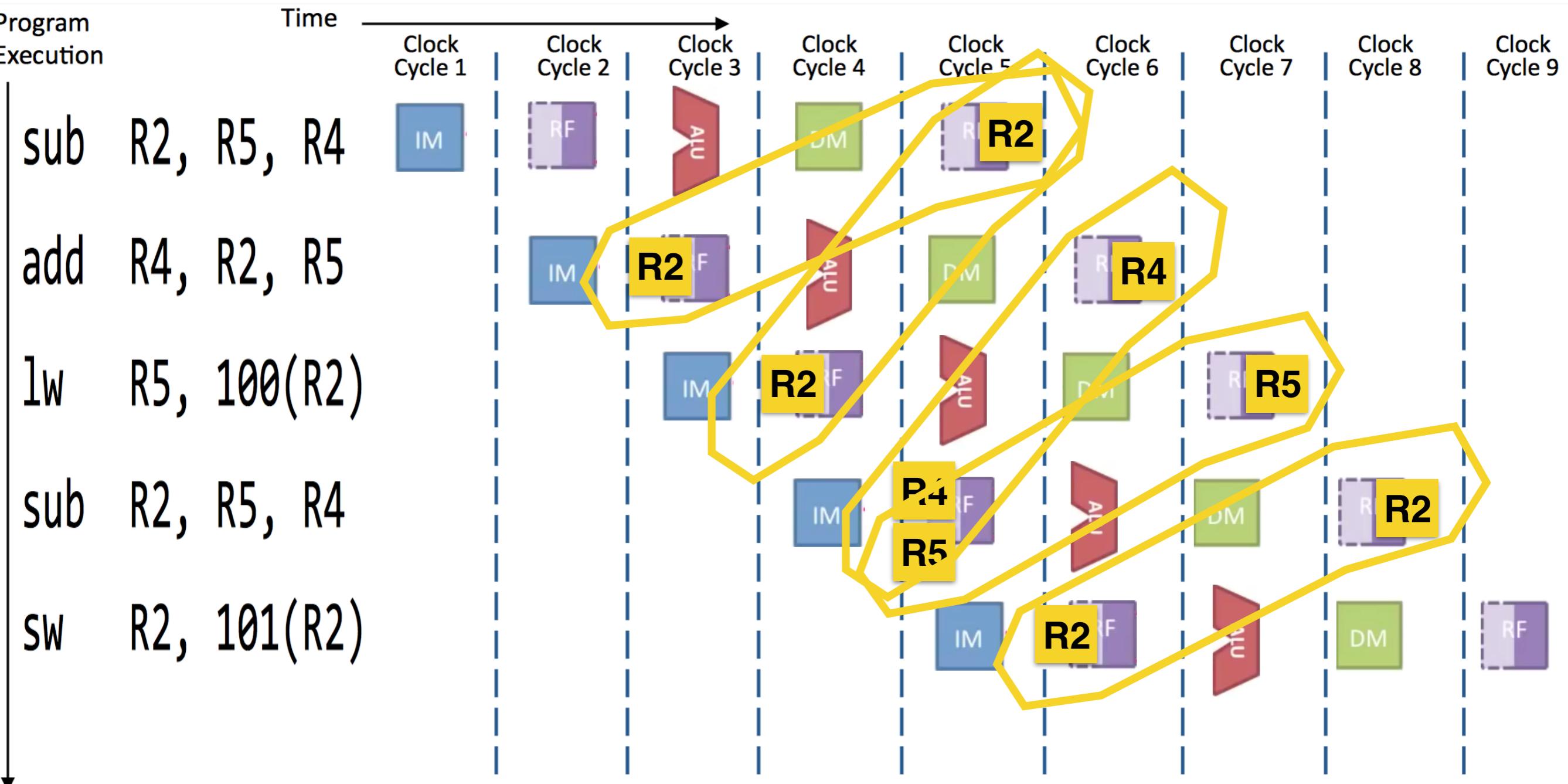
add R4, R2, R5

lw R5, 100(R2)

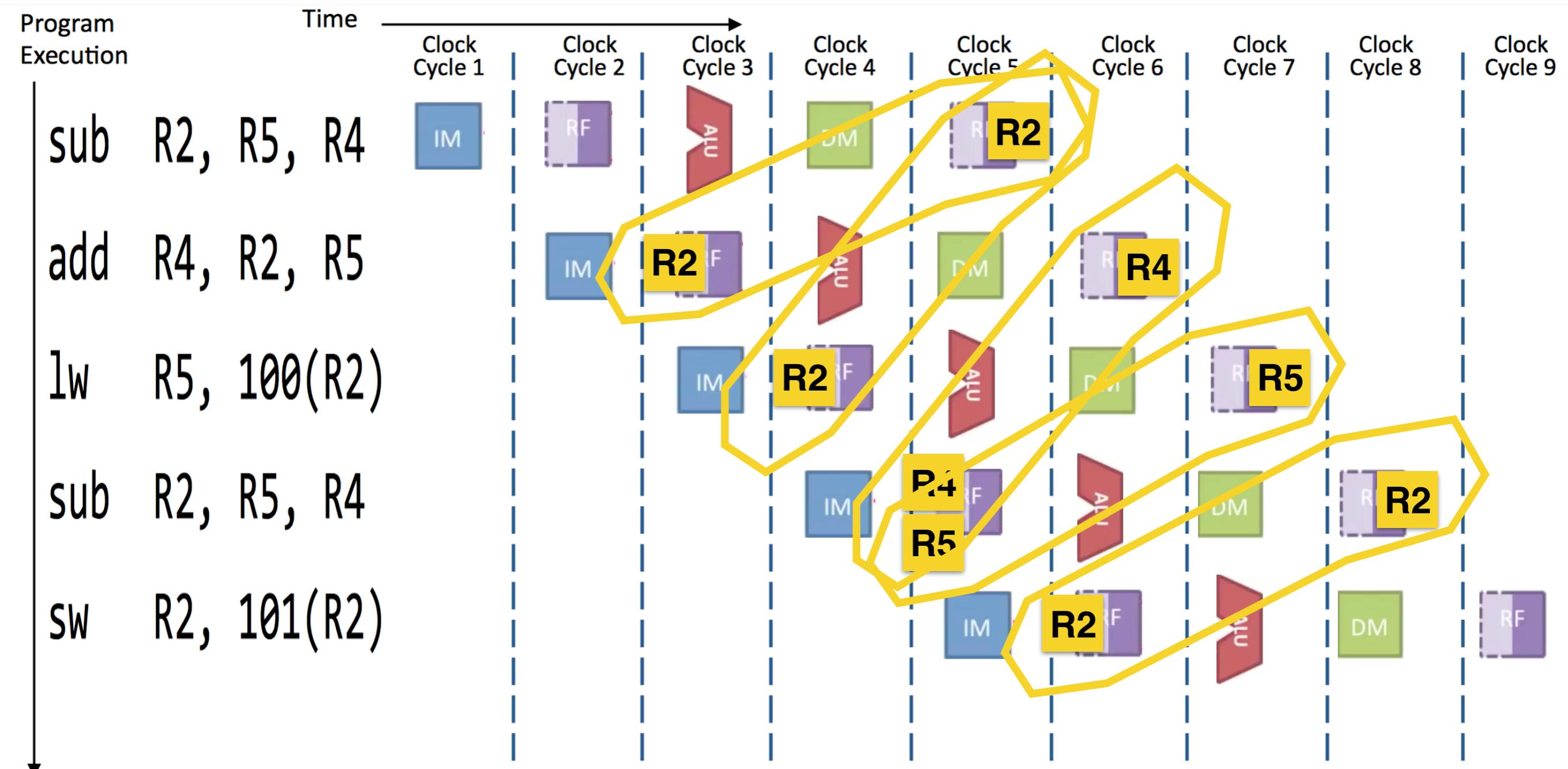
sub R2, R5, R4

sw R2, 101(R2)





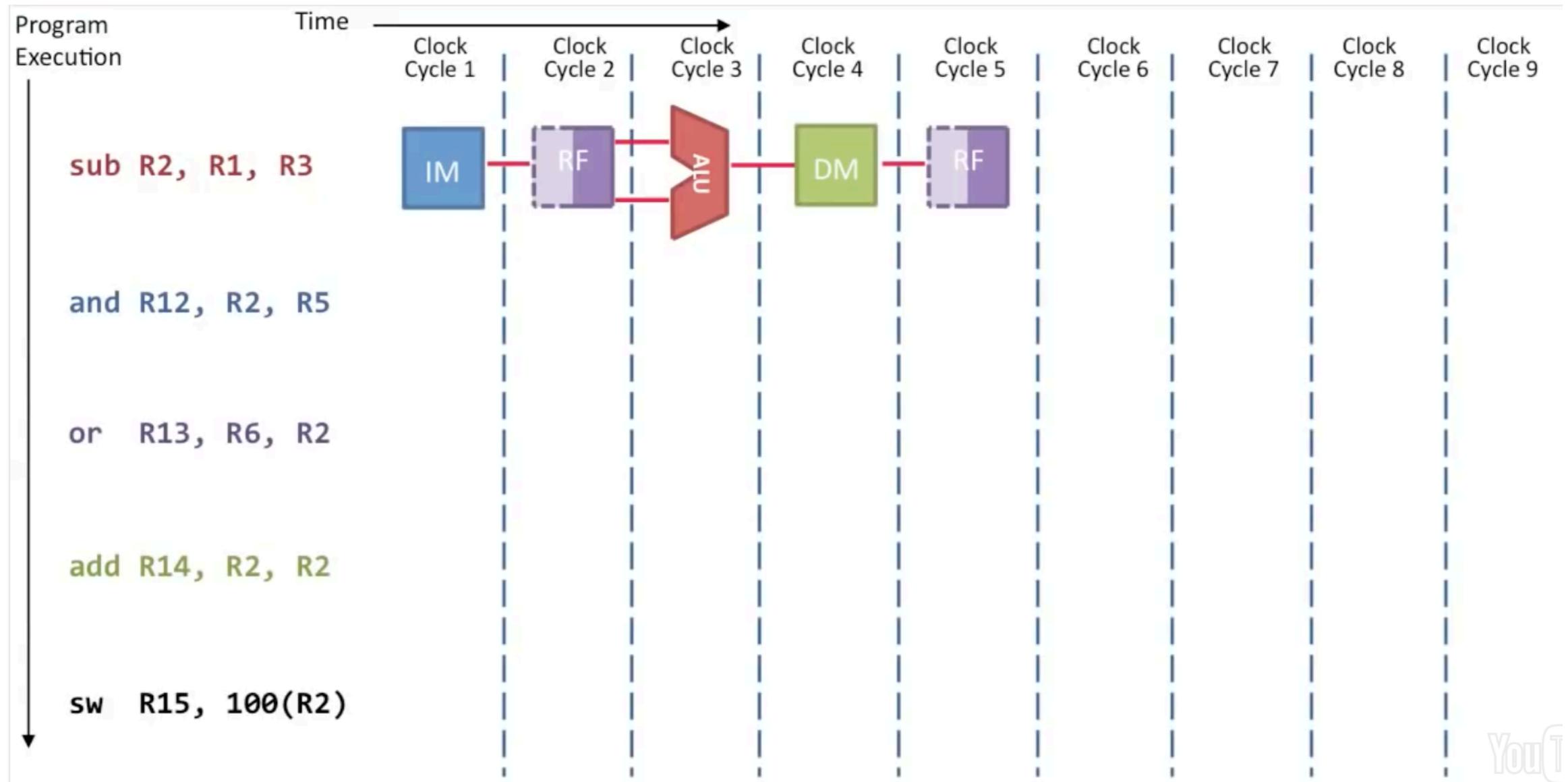
There are 5 data hazards



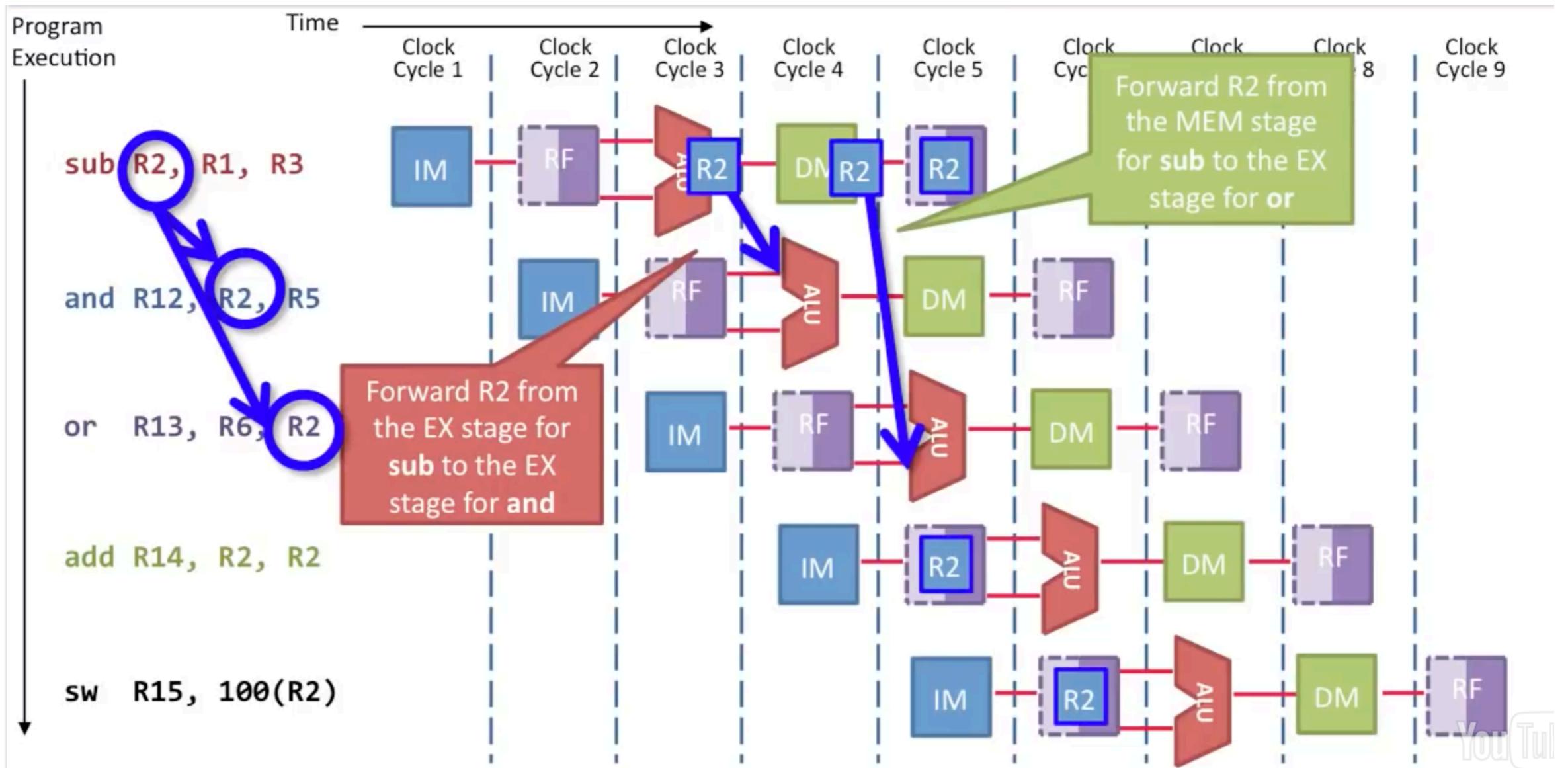
Question: What percentage full is the pipeline in this example?
Here, bubbles/stalls are included.

- we finish 5 instructions in 1 cycles = $5/11 = \textcolor{red}{45\%}$

Forwarding...

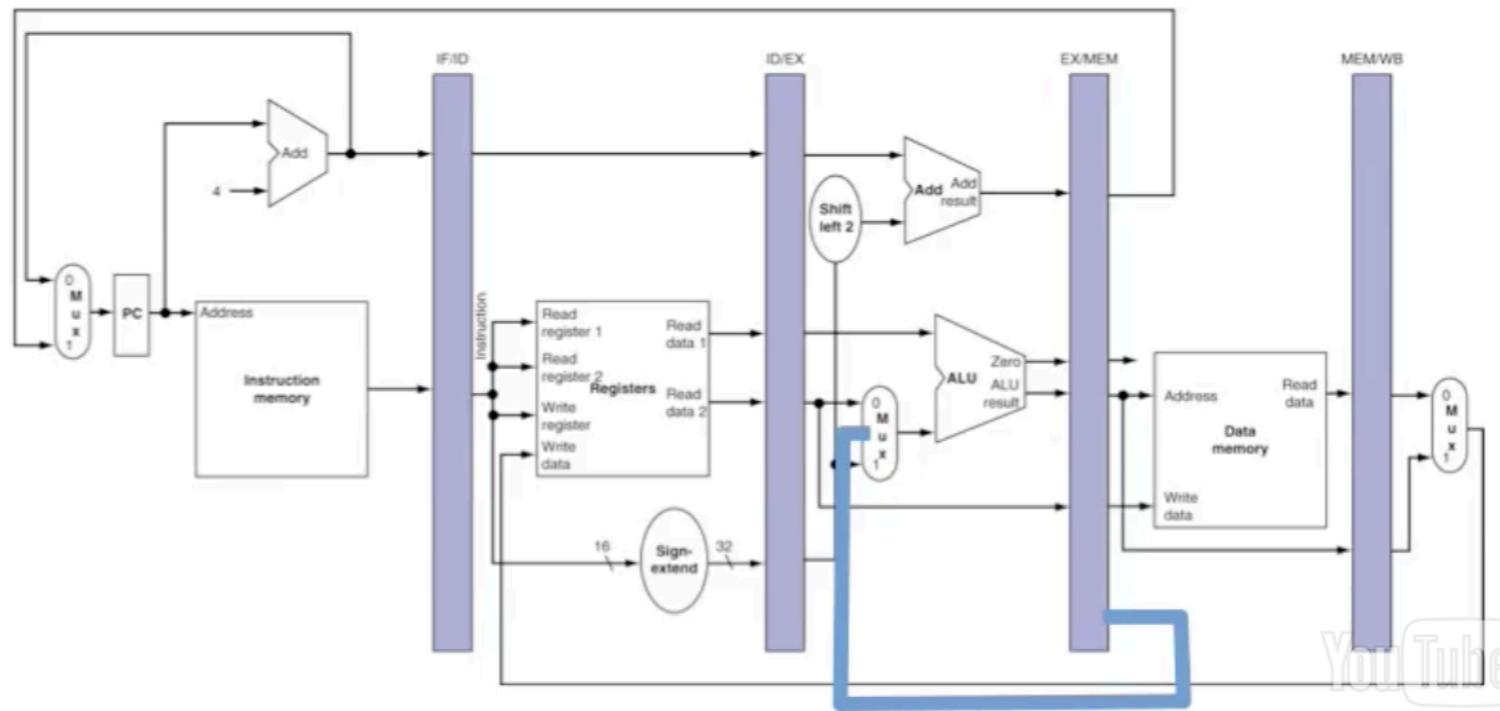


Forwarding...



In order to get rid of stalls, we can use one forwarding path for the following instructions. How many cycles behind is the data forwarded from one stage to another?

```
add R3, R3, R4  
sub R4, R2, R3  
add R4, R4, R4
```



1 cycle behind

Question: A **control hazard** is one in which the choice of the next instruction depends on the results that aren't ready yet. This hazard happens during branch decisions which result to 3 wasted cycles if not resolved. When resolved, the wasted cycle is reduced to only 1 cycle, sometimes called the branch delay slot. *Explain how is control hazard in branch decisions resolved in order to reduce the wasted cycles from 3 to only 1.*

Question: Name three (3) **structural hazards** that exist in a pipelined MIPS processor and tell how they are being resolved.

Note:

- Last topic to watch: **Caches 1 (lecture/quiz Monday 5/15)**
- Final Exam Schedule:
 - **Thursday, 5/25 (1-3PM) Block A**
 - **Thursday 5/25 (3:30-5:30PM) Block B**