Take home assignment – Sam van Kampen

1 a. The following hazards are present:

- Read-After-Write
  - Instruction 1 and 2 (R3)
  - Instruction 3 and 4 (R3)
  - Instruction 2 and 4 (R4)
- Write-After-Read:
  - Instruction 2 and 3 (R3)
  - Instruction 1 and 3 (R3)
- Write-After-Write:
  - Instruction 1 and 3 (R3)

The WAR and WAW hazards are special in the sense that they do not represent a true dependency – the issue is simply that registers are reused, which causes issues when instructions are executed out of order. When we rename registers, the sequence becomes

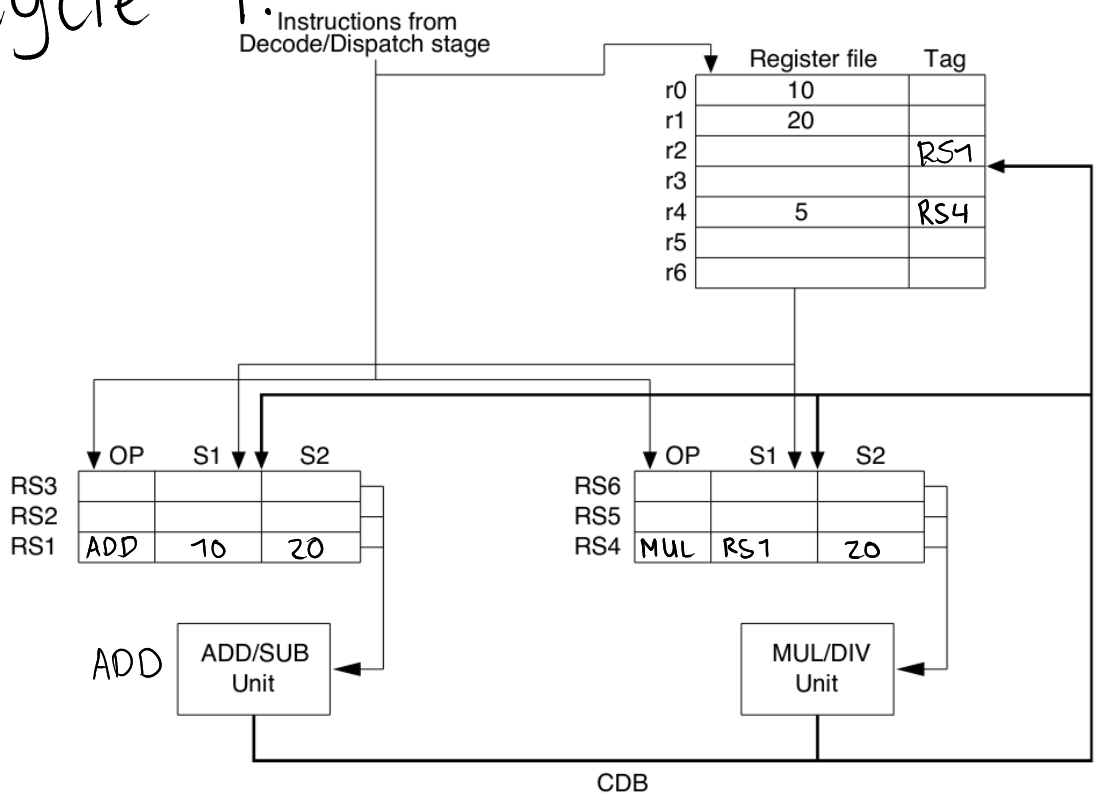
$$
\begin{aligned}
R3' &= R3 \text{ op } R5 \\
R4 &= R3' + 1 \\
R3'' &= R5 + 1 \\
R7 &= R3'' \text{ op } R4
\end{aligned}
$$

and the WAR / WAW dependencies are removed; if we look at instruction 2, for instance, it can no longer read the value produced by instruction 3, even if instruction 3 is executed earlier, because the two registers differ.

16.

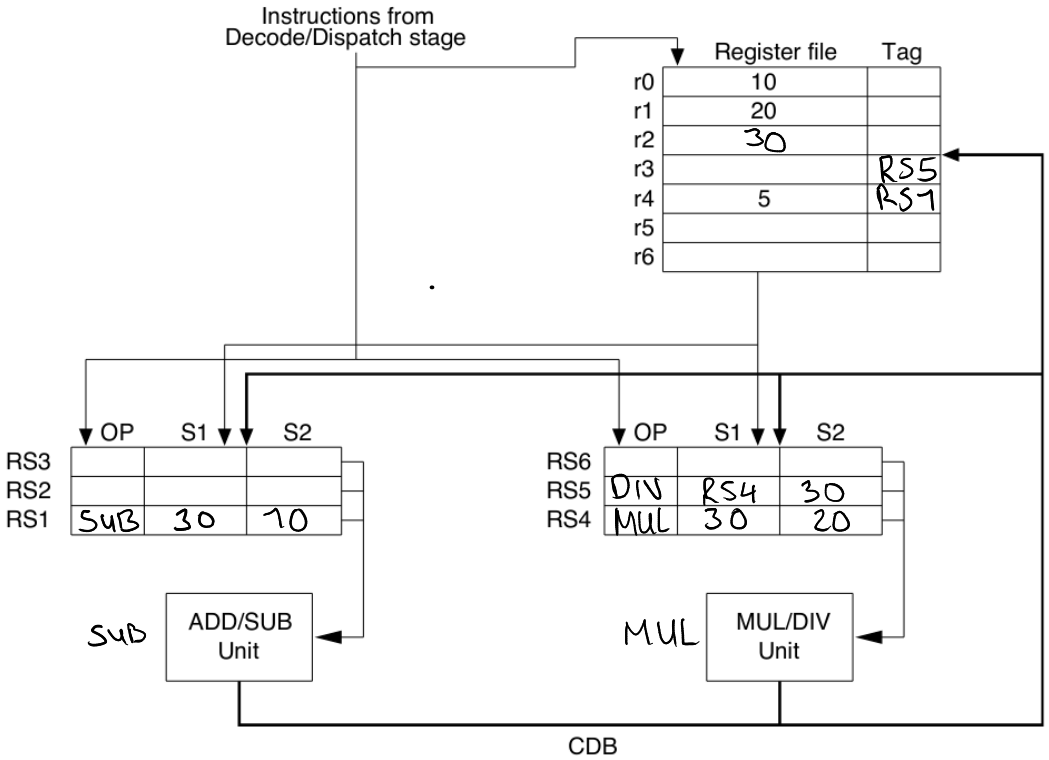
I'm assuming no delay between issue and execution; if the functional unit is free and an instruction enters a RS, I assume it starts execution immediately.

cycle 1:

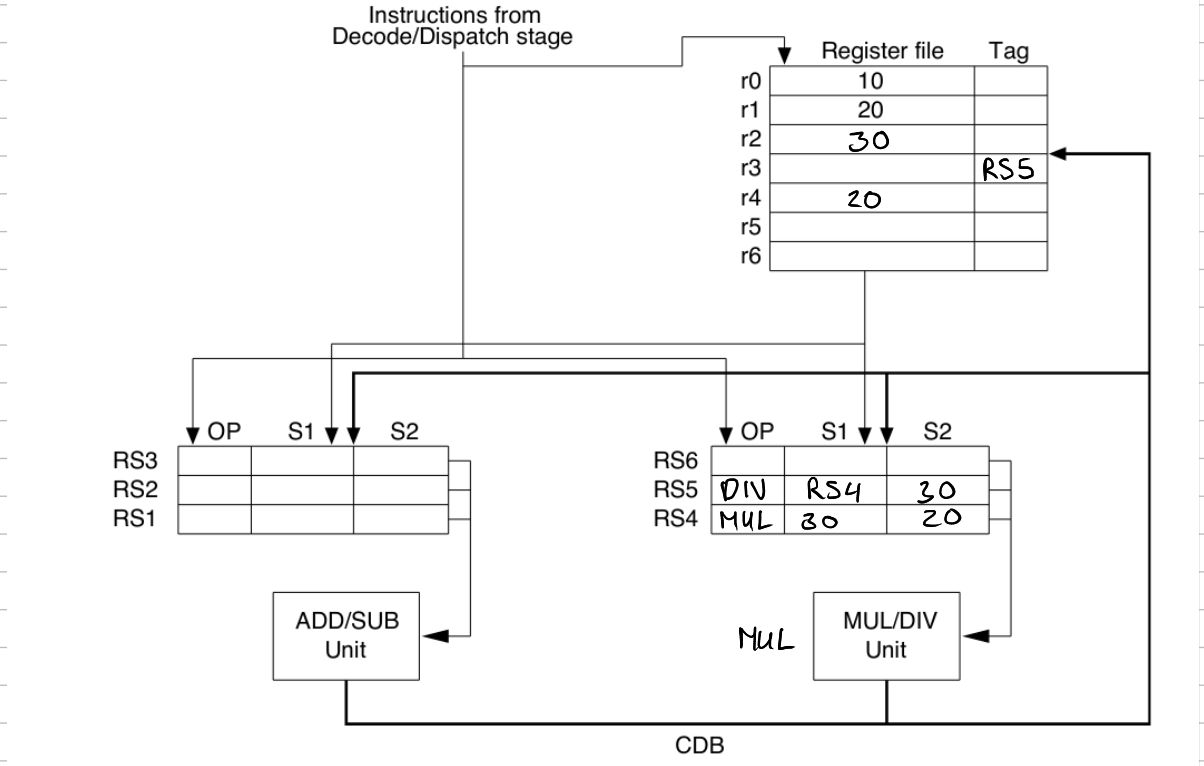


cycle 2: ADD completes, DIV & SUB issued, MUL starts execution.

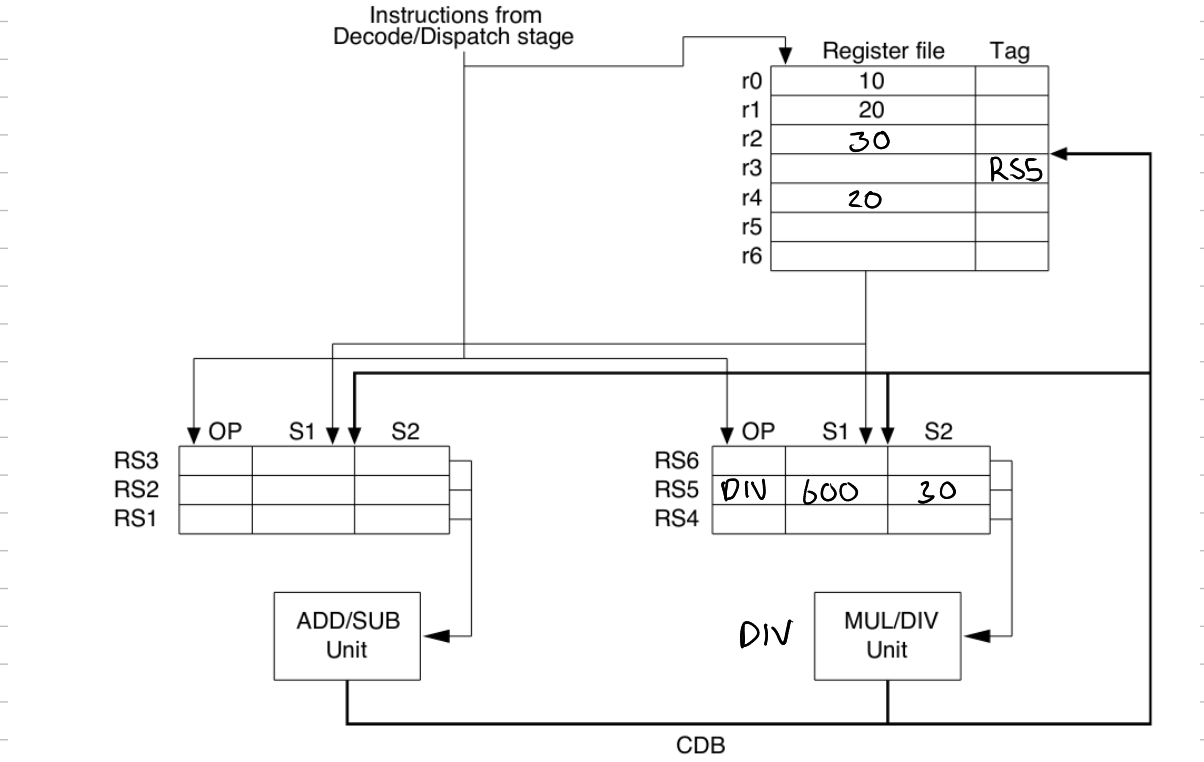
Note that SUB overwrites the tag in the register file.



Cycle 3: SUB completes, writes r4



cycle 7: MUL completes, DIV starts



cycle 12: DIV completes, writes to r3