Hazards:

Data Hazards

Control Hazards

RAW (Rd. after Write) true dependency

WAW (Write after Write) o/p dependency

WAR (Write after Read) anti-dependency

\[ \text{if } P_1 \subseteq S_1 \Rightarrow S_1 \text{ is control dependent on } P_1 \]
\[ \text{if } P_2 \subseteq S_2 \leq R \text{ is C.d. on } P_2 \]

Which of these data hazards need to be preserved and which can be eliminated (by rewriting the program)?

WAW and WAR hazards can be eliminated by using register renaming.

say, R5 then \( WAW \) hazards are eliminated

Add R1, R2, R3
Sub R4, R1, R2
Add R5, R6, R3

Make sure that any use of renamed register is appropriately renamed.
Dynamic Scheduling Usiying Tumaseulo's Algorithm

- Does register renaming dynamically in hardware.

From instr unit

Issued: inorder
but execution is out-of-order & completion is also out-of-order

Ld/St op

Instr queue

Register File

Floating pt. ops

Addrs Unit

Str/Wr

Ld/Wr

Reservatin Station

FP adder

FP multiply

Mem Unit

Common Data Bus

1. Issue - structural hazard
2. Execute
3. Write Result

A: Imm/ effective addr

busy: effective add

op

Q_{i} Q_{k} i_{a}

Rs w