**DATA HAZARDS**

Data hazards are a significant concern in computer organization, particularly in pipelined architectures where multiple instructions are executed concurrently. These hazards occur when there is a dependency between instructions that can result in incorrect execution or stalls in the pipeline. Here are the main types of data hazards:

1. **Read-After-Write (RAW) Hazard**: This occurs when an instruction tries to read a register before a prior instruction writes to it. For example:

*ADD R1,R2,R3;R1 R2 R3*

*SUB R4,R1,R5;R4 R1 R5*

**Write-After-Read (WAR) Hazard**: This occurs when an instruction writes to a register that a later instruction reads from. For example:

*SUB R1,R2,R3 ;R1 R2 R3*

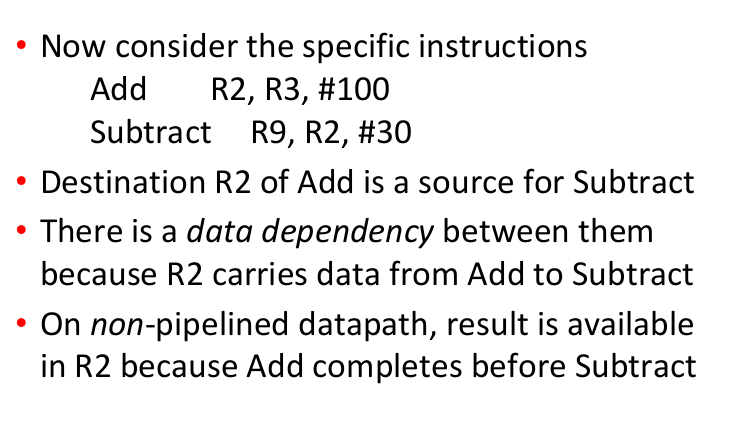
*ADD R4,R1,R5 ; R4 R1 R5*

**Write-After-Write (WAW) Hazard**: This occurs when two instructions write to the same register and the order of execution matters. For example:

*ADD R1,R2,R3;R1 R2 R3*

*SUB R1,R4,R5;R1 R4 R5*

**DATA DEPENDENCIES:**

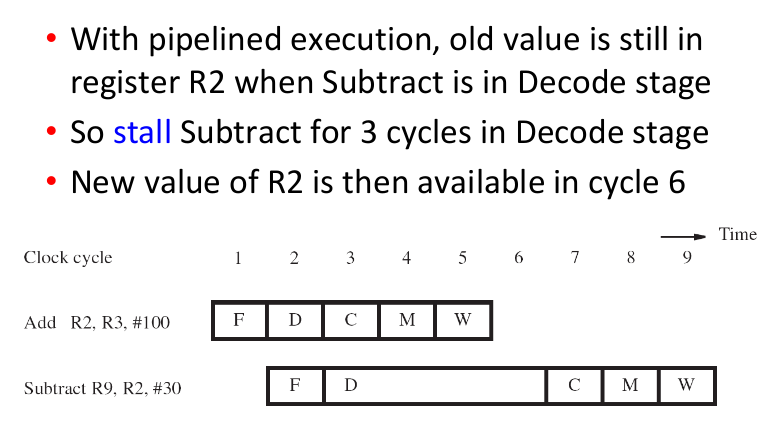
****

**Overcome: Data Hazards**

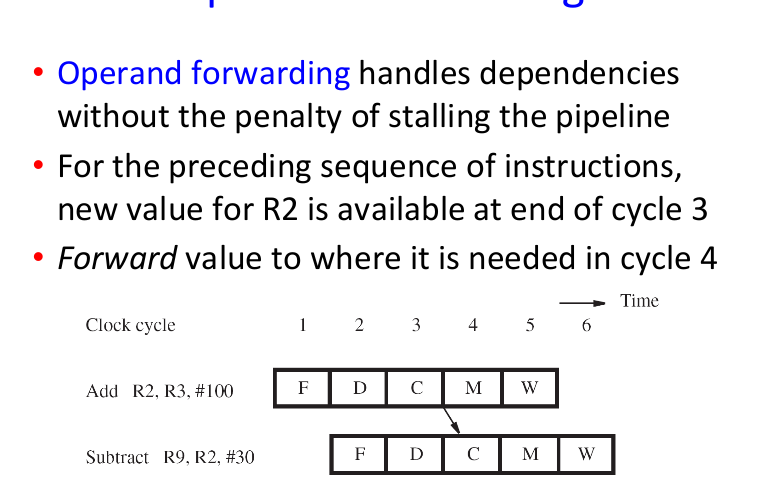
* *1.Stalling the Pipeline*
* *2.Operand Forwarding*
* *3. Software approaches*

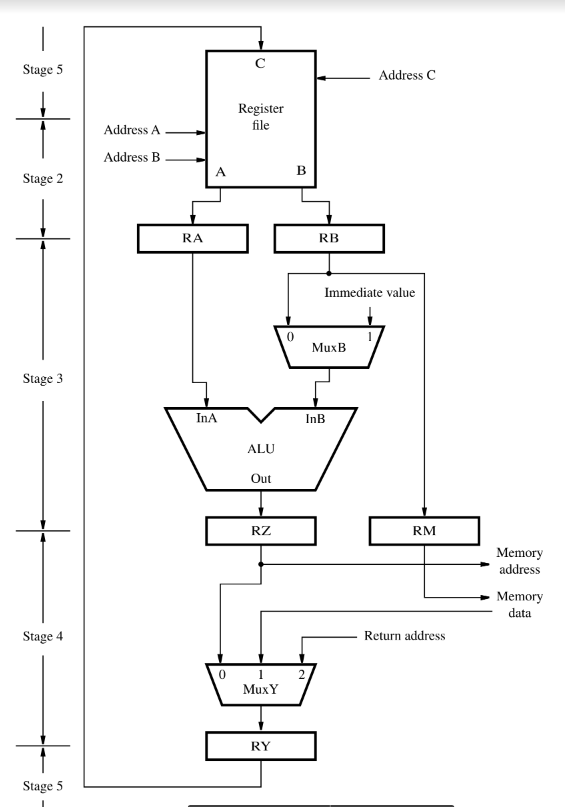
*This first two approach is hardware approach*

***STALLING THE PIPELINE***

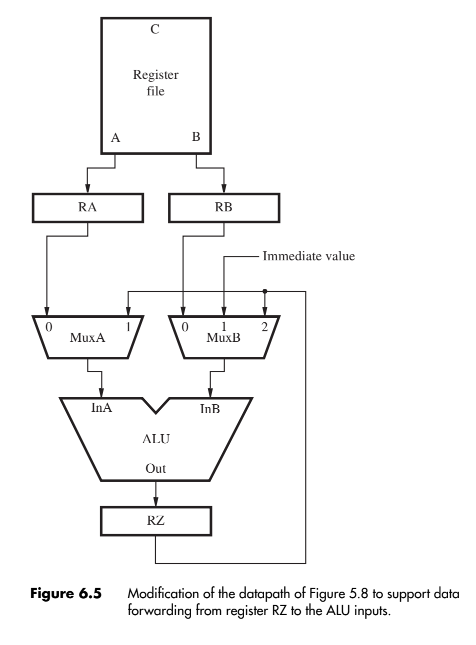
**

*OPERAND FORWARDING*

**

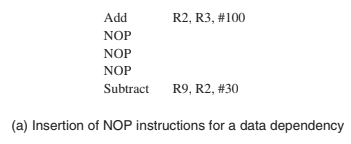
**

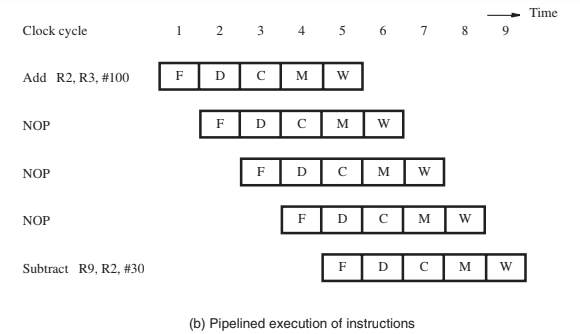
*AFTER THE MODIFICATION OF DATAPATH*

**

*Handling Data Dependencies in Software*

* *The compiler identifies the data dependency between two successive instruction Ij and Ij+1. And insert three explicit NOP instruction between them.*
* *Delay ensures new value available in register but causes total execution time to increase*

**

**

* *Compiler inserting NOPs simplifies hardware.*
* *Code size increases, execution time not reduced.*
* *Compiler can optimize by moving instructions to NOP slots.*
* *Constraints: data dependencies limit NOP slot usage.*