**Data Hazards**

Suppose we are computing the sum of two sensor readings stored in registers:

AR\_ADD R1, R2, R3 # R1 = R2 + R3

AR\_SUB R4, R1, R5 # R4 = R1 - R5

AR\_MUL R6, R4, R7 # R6 = R4 \* R7

Here, R2, R3, R5, and R7 already hold values from memory.

**Pipeline Stages**

The 5-stage pipeline is:

1. **IF**: Instruction Fetch
2. **ID**: Instruction Decode & register read
3. **EX**: Execute (ALU operation)
4. **MEM**: Memory access (if needed)
5. **WB**: Writeback to register

**Step 1: Identify Data Hazards**

1. **AR\_SUB R4, R1, R5** depends on **R1**, which is written by AR\_ADD.
   * Hazard: R1 is not yet available in ID stage when AR\_SUB is decoding.
2. **AR\_MUL R6, R4, R7** depends on **R4**, which is written by AR\_SUB.
   * Hazard: R4 is not yet ready in EX stage when AR\_MUL executes.

This is called a **Read After Write (RAW) hazard**.

**Step 2: Show Pipeline Without Forwarding**

| **Cycle** | **IF** | **ID** | **EX** | **MEM** | **WB** |
| --- | --- | --- | --- | --- | --- |
| 1 | AR\_ADD |  |  |  |  |
| 2 | AR\_SUB | AR\_ADD |  |  |  |
| 3 | AR\_MUL | AR\_SUB | AR\_ADD |  |  |
| 4 |  | AR\_MUL | AR\_SUB | AR\_ADD |  |
| 5 |  |  | AR\_MUL | AR\_SUB | AR\_ADD |
| 6 |  |  |  | AR\_MUL | AR\_SUB |
| 7 |  |  |  |  | AR\_MUL |

* Notice: AR\_SUB reads R1 before AR\_ADD writes it → **pipeline stall needed**.
* Similarly, AR\_MUL depends on R4 → another stall.

This wastes cycles.

**Step 3: Forwarding Solution**

**Forwarding (bypassing)** sends the result from **EX/MEM or MEM/WB stages directly to the ALU input**, so dependent instructions don’t wait until WB stage.

* Forward **R1** from AR\_ADD EX/MEM to AR\_SUB EX stage.
* Forward **R4** from AR\_SUB EX/MEM to AR\_MUL EX stage.

**Step 4: Pipeline With Forwarding**

| **Cycle** | **IF** | **ID** | **EX** | **MEM** | **WB** |
| --- | --- | --- | --- | --- | --- |
| 1 | AR\_ADD |  |  |  |  |
| 2 | AR\_SUB | AR\_ADD |  |  |  |
| 3 | AR\_MUL | AR\_SUB | AR\_ADD |  |  |
| 4 |  | AR\_MUL | AR\_SUB\* | AR\_ADD |  |
| 5 |  |  | AR\_MUL\* | AR\_SUB | AR\_ADD |
| 6 |  |  |  | AR\_MUL | AR\_SUB |
| 7 |  |  |  |  | AR\_MUL |

\*EX stage uses **forwarded values** instead of waiting for WB.