**Hazard estrutural:** quando dois elementos necessitam de aceder ao mesmo hardware.

Situações e solução:

* Aceder a memoria para ler dados e instruções (mete-se memoria da data e das instruções separada)

**Hazards de controlo:** quando uma vais fazer fetch de uma instrução e existe uma instrução que altera o fluxo (Branch ou jump por exemplo) num outro estágio mais avançado. e

Solução:

* calculo do BTA e comparação dos operandos passa a ser feita no estágio 2 do pipeline (ID).
* **Stailing**: atrasa a entrada do pipeline da próxima instrução até saber o resultado do branch condicional. (mau para o tempo)
* **Previsão (prediction):** assume-se que o branch é falso e a próxima instrução a ser executada é o PC+4, se a previsão falharé descartada convertida em nop. Há vários tipos de previsores
* **Delayed branch: d** executa sempre a instrução que se segue ao branch. É implementada com ajuda do assembler que vai trocar as instruções de branch com a instrução anterior desde que não haja dependência entre as duas. Se houver não há troca de instruções e faz um nop.

**Hazards de dados:** dependência entre o resultado de uma instrução e o operando da outra instrução.

* **Stall do pipeline:** parar a progressão da instrução que necessita do resultado na etapa (ID) até a outra chegar na WB.
* **Forwarding ou bypassing:** passar o resultado que se realizou na fase 3 (EX) mais cedo sem esperar pela WB.
* **Reorganização das instruções**: