**Branch**

**Increment I**

1- in R1

2- in R2

3- in R3

4- in R4

5- in R6

6- in R7

7- Push R4

8- JMP R1

8- INC R7

.ORG 30

7- AND R1,R5,R5

8- JZ R2

9- INC R7 –> hazard 1

.ORG 50

10- JZ R3

11- NOT R5

12- INC R5 🡪 hazard 2

13- in R6

14- JZ R6 🡪 hazard 3

15- INC R1 🡪 hazard 4

.ORG 100

16- ADD R0,R0,R0

17- out R6

18- rti

.ORG 200

19- POP R6

20- Call R6 🡪 hazard 5

21- INC R6

22- NOP

23- NOP

.ORG 300

24- Add R3,R6,R6

25- Add R1,R2,R1

25- ret

26- INC R7

.ORG 500

27- NOP

28- NOP

Hazard1:

Control hazard; as JZ will be executed in execute stage, So INC instruction will be in decode stage and it is falsely executed.

Hazard2:

Data Hazard on R5, need 2 NOP instructions.

Hazard3:

Data Hazard on R6, need 2 NOP instructions.

Hazard4:

Control Hazard, as Hazard 1

Hazard5:

Data Hazard on R6, needs 3 Nop instructions.

**Screenshots:**

**Increment II**All data hazards are supposedly solved by the forwarding unit. only hazards still exist are:

Hazard 1, 4

And hazard 5 need 2 NOP operation

**Increment III**

Hazard 5 is solved by hazard detection unit in fetch stage for branching instructions.

**Increment IV**

Hazard 1, 4 are solved by adding dynamic branch prediction.

**Screenshots:**