4.13.1 If there is no forwarding or Hazard detection, insert nops to ensure correct execution.

```
ADD   R5, R2, R1
 NOP
 NOP
 LW   R3, 4 (R5)
 LW   R2, 0 (R2)
 NOP
 OR   R3, R5, R3
 NOP
 NOP
 SW   R3, 0(R5)
```

4.13.2 Repeat 4.13.1 but now use nops only when a hazard cannot be avoided by changing or rearranging these instructions. You can assume register R7 can be to hold temporary values in your modified code.

Ans) We can use R7 to eliminate WAR/WAW dependences.
We can move by swapping instruction
with other no dependency instructions

```
I1: Add R5, R2, R1

I3 : LW R2, 0(R2)

NOP
I2: LW R3, 4(R5)
NOP
NOP
I4: OR R3, R5, R3
NOP
NOP
I5: SW R3, 0(R5)
```

4.13.3 If ~~professor~~ processor has forwarding but we forgot to implement the hazard detection unit, what happens when this code executes.

Ans> With forwarding, the hazard detection unit is still needed because it must insert a one-cyle stall whenever the load supplies a value to the instruction that follows that load.

The ~~de~~ instruction depends on immediately preceding load gets the stale value the register had before the load instruction, without the hazard detection.

4.13.4 If there is forwarding for the first five cycles during the exowtion of this code specify which signals are asserted in each cycle by hazard detection & forwarding units

Ans)

| Instructions | First five cycles 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Add R5, R2, R1 | IF | ID | Ex | MEM | WB |
| LW R3, 4(R5) | | IF | ID | Ex | MEM |
| LW R2, 0(R2) | | | IF | ID | Ex |
| OR R3, R5, R3 | | | | IF | ID |
| SW R3, 0(R5) | | | | | IF |

| | | ALU | ALU2 |
|---|---|---|---|
| 1: | PCWrite = 1 | X | X |
| 2: | 1 | x | x |
| 3: | 1 | 0 | 0 |
| 4: | 1 | 1 | 0 |
| 5: | 1 | 0 | 0 |

The outputs are PC write, IF/IDwrite & ID/EXZero (which controls the Mux after the output of the Control unit)

~~Also~~

The outputs of forwarding unit is ALUin1 & ALUin2, which controls Muxes that select the first & second input of the ALU.

4.13.5 If there is no forwarding what new inputs & output signals do we need for the hazard detection unit in Fig. 4.60? Using this instruction sequence as an example explain why each signal is needed.

Ans) The instruction in ID stage currently needs to be stalled if it depends on a value produced by the instruction in the EX or the instruction in MEM stage.

We check destination reg of these.

. We need to Check Rd for R-type insTn
For Ex stage ~~we need~~ need ~~to take~~

For MEM stage the destination reg is already pulled

The additional input to hazard ~~input~~ detection unit are Rd from the Id/EX pipeline register & the output number of the output register from the EX/MEM reg.

No additional pipline is needed

13.6 For new hazard detection unit from 4.13.5 specify which output signals it asserts in each of the first five cycles during the execution of this code.

| Instructions | First five cycles | | | | | Signals |
|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | |
| Add R5, R2, R1 | IF | ID | EX | MEM | WB | 1 : PCWrite=1 |
| LW R3, 4(R5) | | IF | ID | *** | *** | 2 : PCWrite=1 |
| LW R2, 0(R2) | | | IF | *** | *** | 3 : PCWrite=1 |
| OR R3, R5, R3 | | | | | *** | 4 : PCWrite=0 |
| SW R3, 0(R5) | | | | | | 5 : " = 0 |