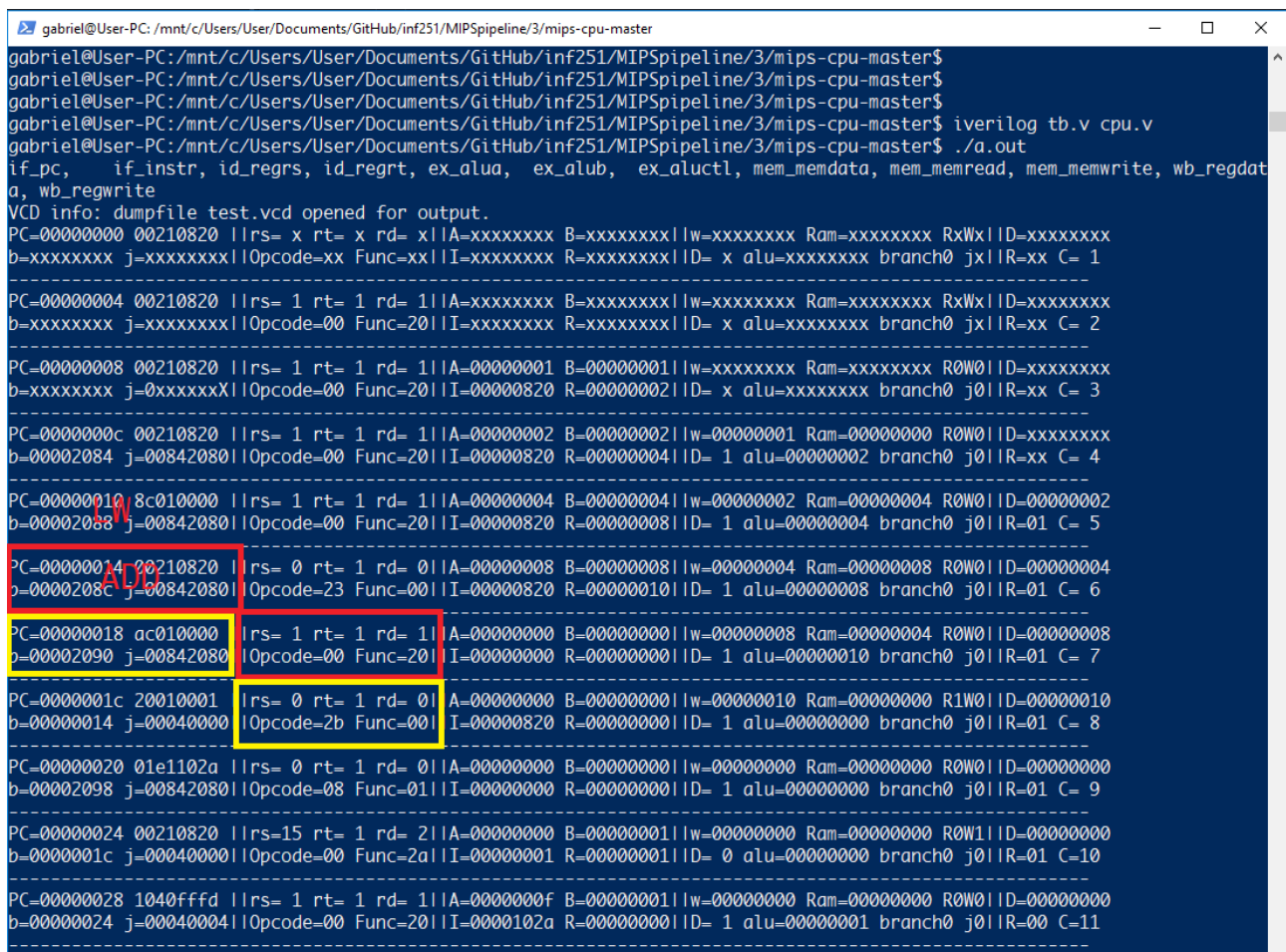


DOCUMENTAÇÃO

Para remover o hazard, apenas definimos que nunca ocorrerá o stall_s1_s2.

```
always @(*) begin
    if (memread_s3 == 1'b1 && ((rt == rt_s3) || (rs == rt_s3)) ) begin
        stall_s1_s2 <= 1'b0; // no stall
    end else
        stall_s1_s2 <= 1'b0; // no stall
end
```

Com isso, mesmo que a caixinha de hazard detecte esse caso, ela não irá alterar a sequencia do pipeline, simulando assim a remoção/ausencia do hazard.



```
gabriel@User-PC: /mnt/c/Users/User/Documents/GitHub/inf251/MIPSPipeline/3/mips-cpu-master$
gabriel@User-PC: /mnt/c/Users/User/Documents/GitHub/inf251/MIPSPipeline/3/mips-cpu-master$
gabriel@User-PC: /mnt/c/Users/User/Documents/GitHub/inf251/MIPSPipeline/3/mips-cpu-master$ iverilog tb.v cpu.v
gabriel@User-PC: /mnt/c/Users/User/Documents/GitHub/inf251/MIPSPipeline/3/mips-cpu-master$ ./a.out
if_pc, if_instr, id_regrs, id_regrt, ex_alua, ex_alub, ex_aluctl, mem_memdata, mem_memread, mem_memwrite, wb_regdat
a, wb_regwrite
VCD info: dumpfile test.vcd opened for output.
PC=00000000 00210820 |lrs= x rt= x rd= x |IA=xxxxxxx B=xxxxxxx |lw=xxxxxxx Ram=xxxxxxx RxWx|ID=xxxxxxx
b=xxxxxxx j=xxxxxxx |lopcode=xx Func=xx |I=xxxxxxx R=xxxxxxx |ID= x alu=xxxxxxx branch0 j0 |IR=xx C= 1
-----
PC=00000004 00210820 |lrs= 1 rt= 1 rd= 1 |IA=xxxxxxx B=xxxxxxx |lw=xxxxxxx Ram=xxxxxxx RxWx|ID=xxxxxxx
b=xxxxxxx j=xxxxxxx |lopcode=00 Func=20 |I=xxxxxxx R=xxxxxxx |ID= x alu=xxxxxxx branch0 j0 |IR=xx C= 2
-----
PC=00000008 00210820 |lrs= 1 rt= 1 rd= 1 |IA=00000001 B=00000001 |lw=xxxxxxx Ram=xxxxxxx R0W0 |ID=xxxxxxx
b=xxxxxxx j=0xxxxxx |lopcode=00 Func=20 |I=00000820 R=00000002 |ID= x alu=xxxxxxx branch0 j0 |IR=xx C= 3
-----
PC=0000000c 00210820 |lrs= 1 rt= 1 rd= 1 |IA=00000002 B=00000002 |lw=00000001 Ram=00000000 R0W0 |ID=xxxxxxx
b=00002084 j=00842080 |lopcode=00 Func=20 |I=00000820 R=00000004 |ID= 1 alu=00000002 branch0 j0 |IR=xx C= 4
-----
PC=00000010 00210820 |lrs= 1 rt= 1 rd= 1 |IA=00000004 B=00000004 |lw=00000002 Ram=00000004 R0W0 |ID=00000002
b=00002088 j=00842080 |lopcode=00 Func=20 |I=00000820 R=00000008 |ID= 1 alu=00000004 branch0 j0 |IR=01 C= 5
-----
PC=00000014 00210820 |lrs= 0 rt= 1 rd= 0 |IA=00000008 B=00000008 |lw=00000004 Ram=00000008 R0W0 |ID=00000004
b=0000208c j=00842080 |lopcode=23 Func=00 |I=00000820 R=00000010 |ID= 1 alu=00000008 branch0 j0 |IR=01 C= 6
-----
PC=00000018 a010000 |lrs= 1 rt= 1 rd= 1 |IA=00000000 B=00000000 |lw=00000008 Ram=00000004 R0W0 |ID=00000008
b=00002090 j=00842080 |lopcode=00 Func=20 |I=00000000 R=00000000 |ID= 1 alu=00000010 branch0 j0 |IR=01 C= 7
-----
PC=0000001c 20010001 |lrs= 0 rt= 1 rd= 0 |IA=00000000 B=00000000 |lw=00000010 Ram=00000000 R1W0 |ID=00000010
b=00000014 j=00040000 |lopcode=2b Func=00 |I=00000820 R=00000000 |ID= 1 alu=00000000 branch0 j0 |IR=01 C= 8
-----
PC=00000020 01e1102a |lrs= 0 rt= 1 rd= 0 |IA=00000000 B=00000000 |lw=00000000 Ram=00000000 R0W0 |ID=00000000
b=00002098 j=00842080 |lopcode=08 Func=01 |I=00000000 R=00000000 |ID= 1 alu=00000000 branch0 j0 |IR=01 C= 9
-----
PC=00000024 00210820 |lrs=15 rt= 1 rd= 2 |IA=00000000 B=00000001 |lw=00000000 Ram=00000000 R0W1 |ID=00000000
b=0000001c j=00040000 |lopcode=00 Func=2a |I=00000001 R=00000001 |ID= 0 alu=00000000 branch0 j0 |IR=01 C=10
-----
PC=00000028 1040fffd |lrs= 1 rt= 1 rd= 1 |IA=0000000f B=00000001 |lw=00000000 Ram=00000000 R0W0 |ID=00000000
b=00000024 j=00040004 |lopcode=00 Func=20 |I=0000102a R=00000000 |ID= 1 alu=00000001 branch0 j0 |IR=00 C=11
-----
```

A partir da imagem podemos concluir que, de fato, não ocorre o stall_s1_s2 pois após a instrução de LW a instrução de ADD não fica travada, e o pipeline continua sua execução normalmente.

Exemplo gerado com o arquivo “addlwsowlbeq.txt”.