数据冒险由两个旁路单元、一个 LW 数据冒险处理单元和一个条件分支数据冒险处理单元解决，两个旁路单元控制的旁路分别位于 ALU 输入端口和条件分支判断单元的输入端口，分别如图 1 和图 2 所示。均旁路 MEN 级 ALU 的计算结果和 WB 级待写回的数据，其中 MEN 级的优先级最高，若与 WB 级和 MEN 级均发生数据冒险则优先旁路 MEN 级的数据。两个旁路控制单元通过该级指令的操作数与 MEN 和 WB 级指令的操作数对比，相同则发生数据冒险，那么就需要旁路回来，具体硬件实现分别如图 4 和图 5，点击主文件中相应单元也可跳转至具体的实现项目文件。

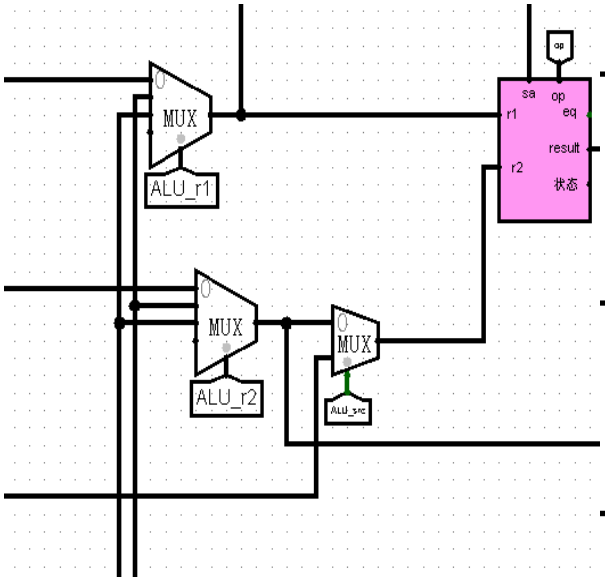

图 1

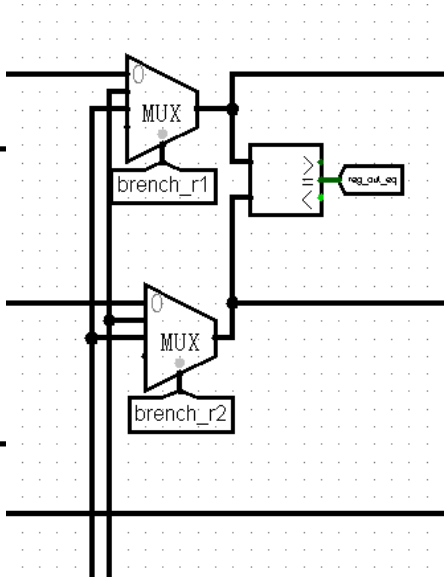

图 2

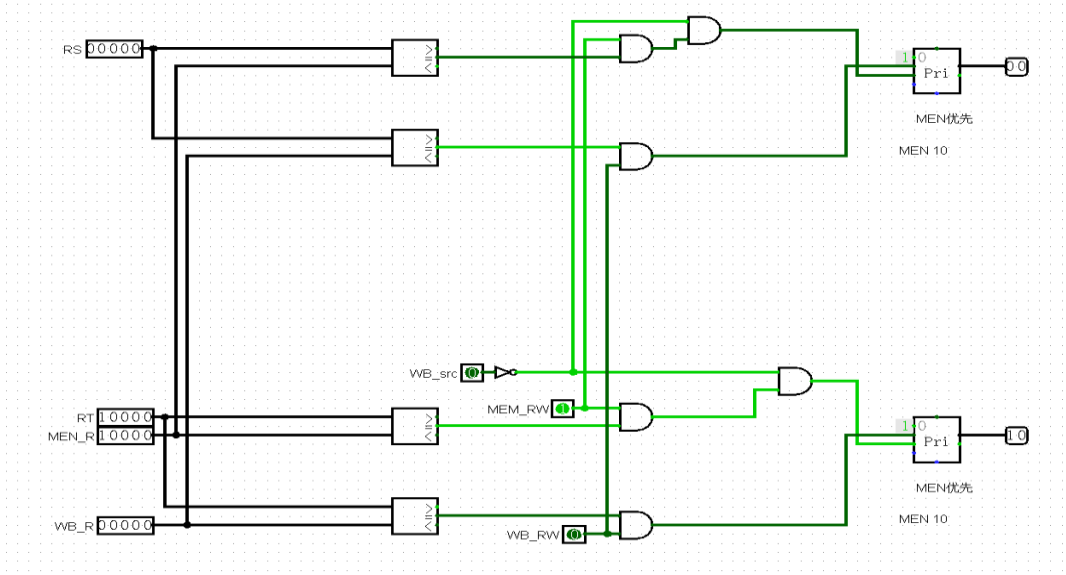

图 3

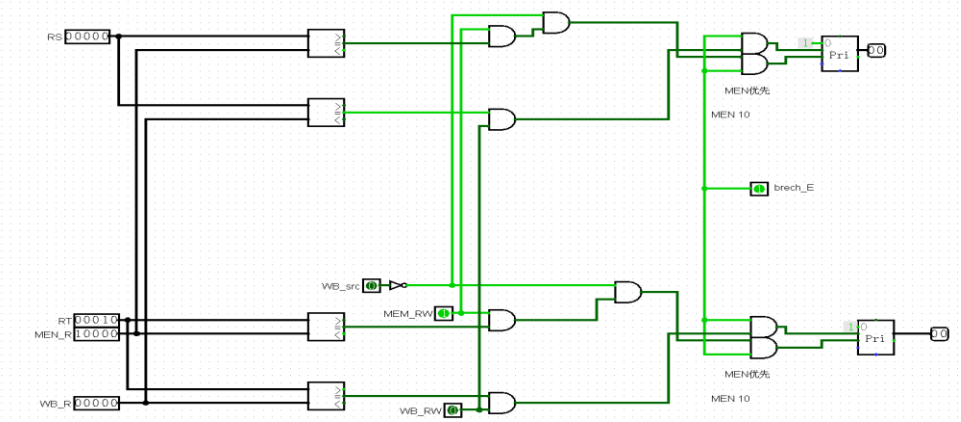

图 4

LW 数据冒险控制单元是处理如下情形的数据冒险：

lw rt offset(base)

add rd rs rt

当 lw 指令的目标寄存器与其后面的第一条指令(不一定是 add)发生 RAW 依赖，如上例 lw 的 rt 被 add 的 rs 或者 rt 使用。那么 add 执行到 EX 级时，由于 MEM 级无法将存储器的输出向 EX 级旁路，lw 从存储器取出的 rt 则无法旁路至 EX 级 ALU 的输入端口，因此 add 指令无法得到正确的操作数。因此 LW 数据冒险控制单元所执行的操作是当 add 执行到 ID 级时，使其暂停一个周期，如图 5 和图 6 所示。冒险处理单元具体执行的操作是将 PC 寄存器和 IF/ID 级间寄存器的写使能置低电平，下一个 clk 的上升沿到来时将 ID/EX 级的寄存器复位端口置高电平，使寄存器清 0，图 7 是冒险处理单元输出的寄存器复位信号，需先经过一个下降沿触发的寄存器锁存，至下一个 clk 到来在输出。

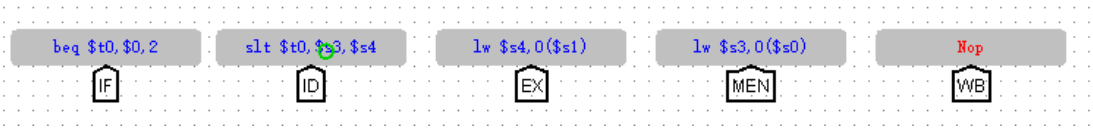

图 5 ID 级的 slt 与 ex 级的 lw 发生 RAW 依赖，产生数据冒险

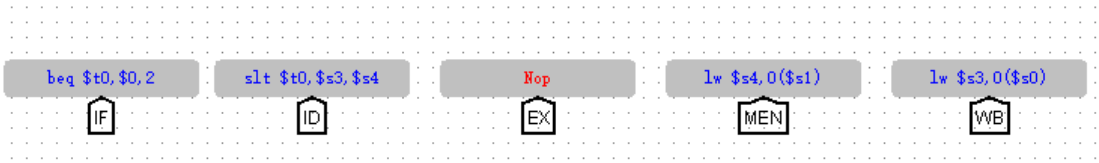

图 6 LW 数据冒险单元处理完数据冒险之后

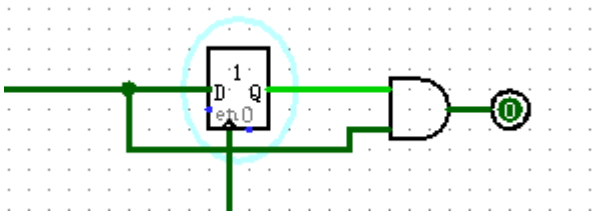

图 7

条件分支数据冒险处理单元是处理如下情形的数据冒险：

add rd rs rt

BEQ/BNE rs rt offset

当一条运算指令的目标寄存器与后面的第一条条件分支指令发生 RAW 依赖，如上例 add 指令的 rd 被 BEQ/BNE 的 rs 或者 rt 使用。那么当条件分支指令执行到 ID 级时，其前一条指令刚刚执行到 EX 级，由于 EX 级的 ALU 结果无法旁路回 ID 级的条件分支判断单元的输入端口，因此在判断条件时无法得到正确的操作数，所以只能将条件分支指令延迟一个周期，待前一天指令执行到 MEN 级，将 ALU 运算结果旁路回来，如图 8 和图 9 所示。冒险处理单元具体执行的操作是将 PC 寄存器和 IF/ID 级间寄存器的写使能置低电平，下一个 clk 的上升沿到来时将 ID/EX 级的寄存器复位端口置高电平，使寄存器清 0，图 10 是冒险处理单元输出的寄存器复位信号，需先经过一个下降沿触发的寄存器锁存，至下一个 clk 到来在输出。

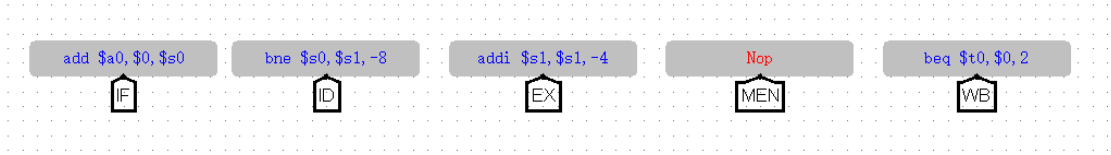

图 8　ID 级的 bne 与 EX 级的 addi 指令发送 RAW 依赖，产生数据冒险

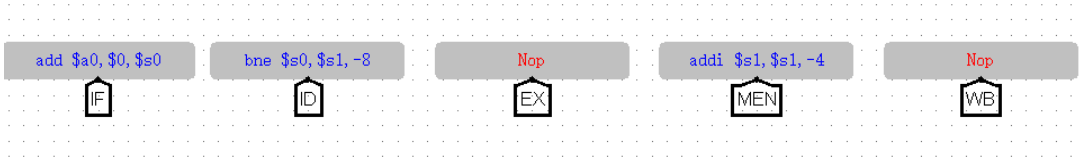

图 9　分支数据冒险单元处理完之后

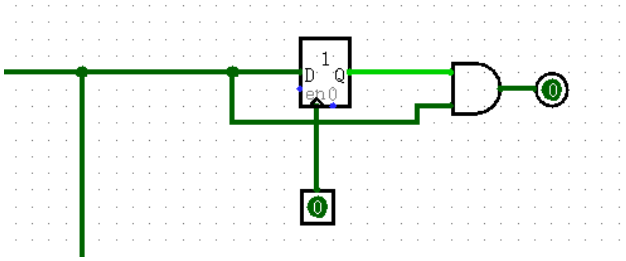

图 10