Data path:
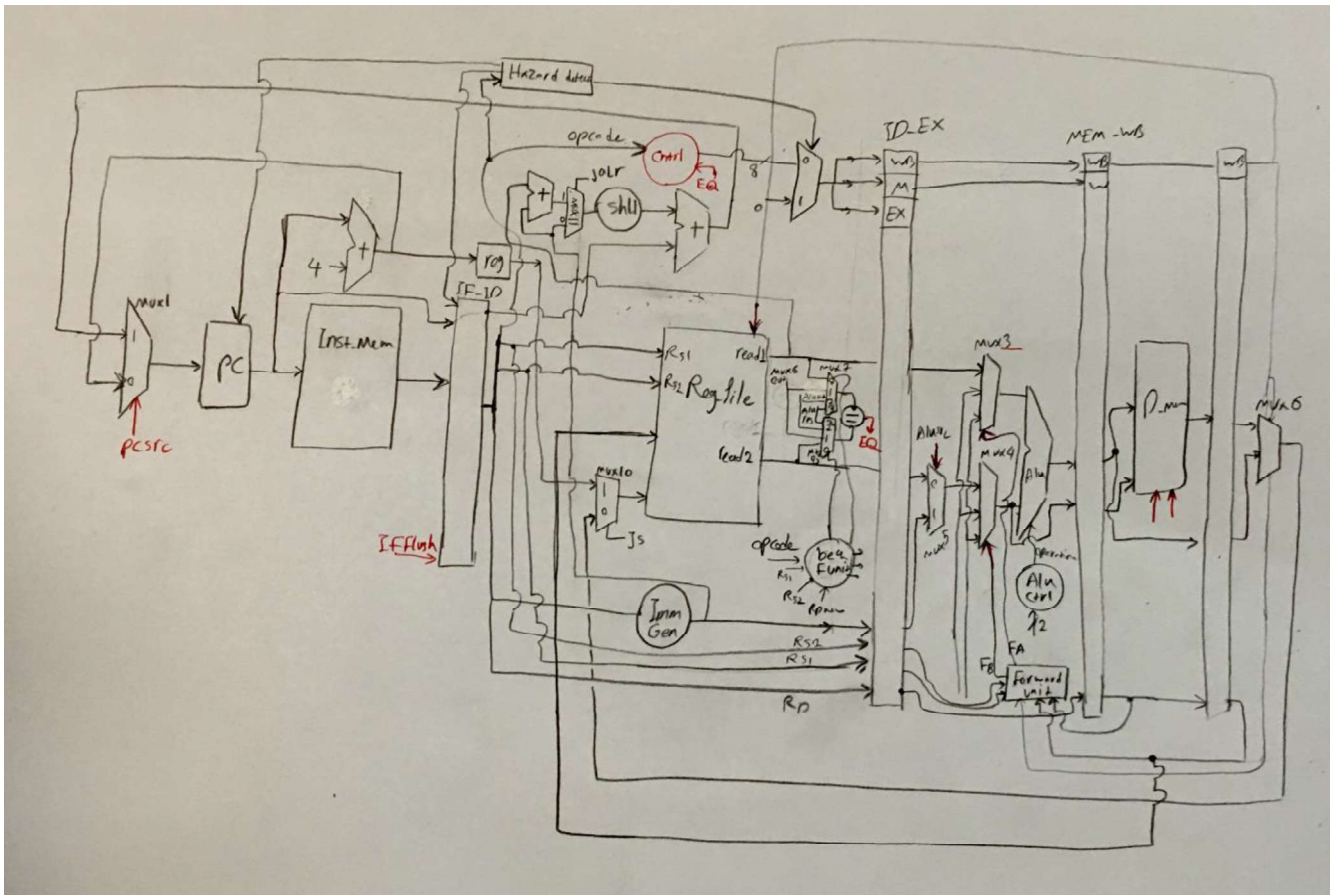
We tested this code that includes any kind of possible hazards:

```
//              add    R8, R0, R0 0
//              addi   R10,  R0, 20 4
//              ld     R2, 1000(R8) 8
//              add    R3, R0, R8     12
// Loop:        beq    R8,  R10, END_LOOP 16
//              add    R13, R8, R8        20      /////
//              add    R13, R13, R13      24      ////
//              add    R13, R13, R13      28
//              ld     R7, 1000(R13)      32          ///
//              slt    R11, R7, R2        36
//              beq    R0, R11, END_IF    40      ////////////////
//              add    R2, R0, R7         44      //////
//              add    R3, R0, R8         48
// END_IF: addi       R8, R8, 1          52
//              J      LOOP               56      //// (beq)
// END_LOOP: sd, R2, 2000(R0)            60
//              sd, R3, 2004(R0)         64
```
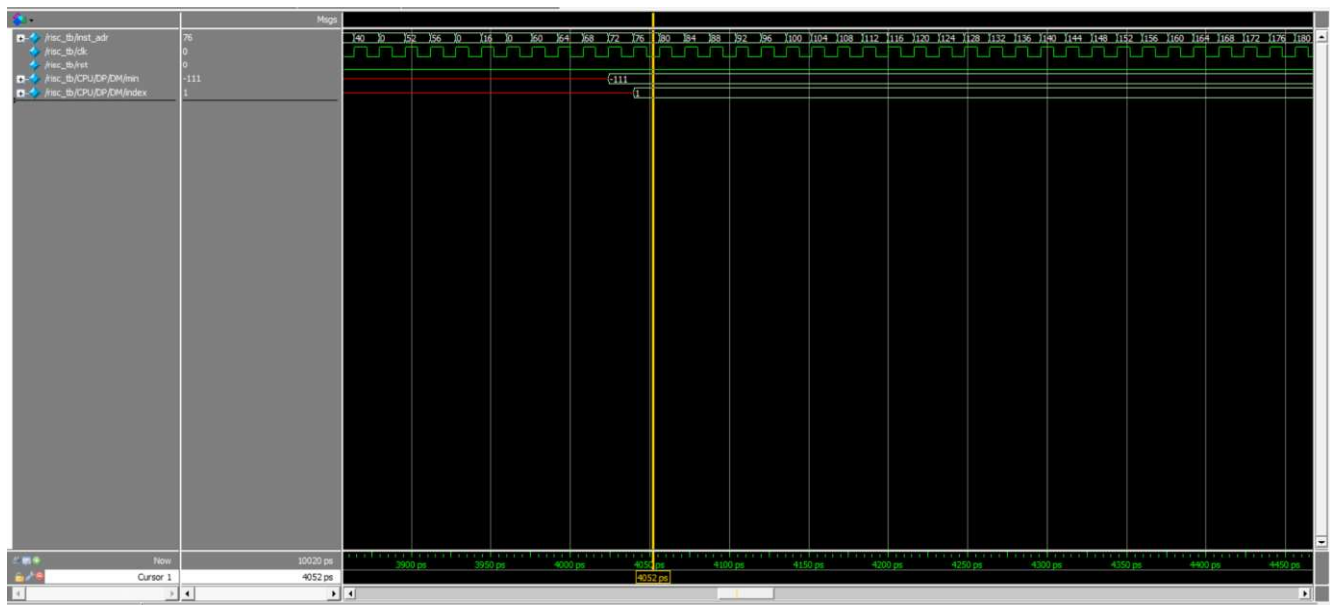
Simulations:


```
64'd8;    //0
64'd111;  //1
64'd11;   //2
64'd5;    //3
64'd4;    //4
64'd5;    //5
64'd6;    //6                Min = 0
64'd7;    //7
64'd7;    //8                Index = 9
64'd0;    //9
64'd101;  //10
64'd11;   //11
64'd12;   //12
64'd13;   //13
64'd5;   //14
64'd14;   //15
64'd14;   //16
64'd11;   //17
64'd11;   //18
64'd11; //19
```
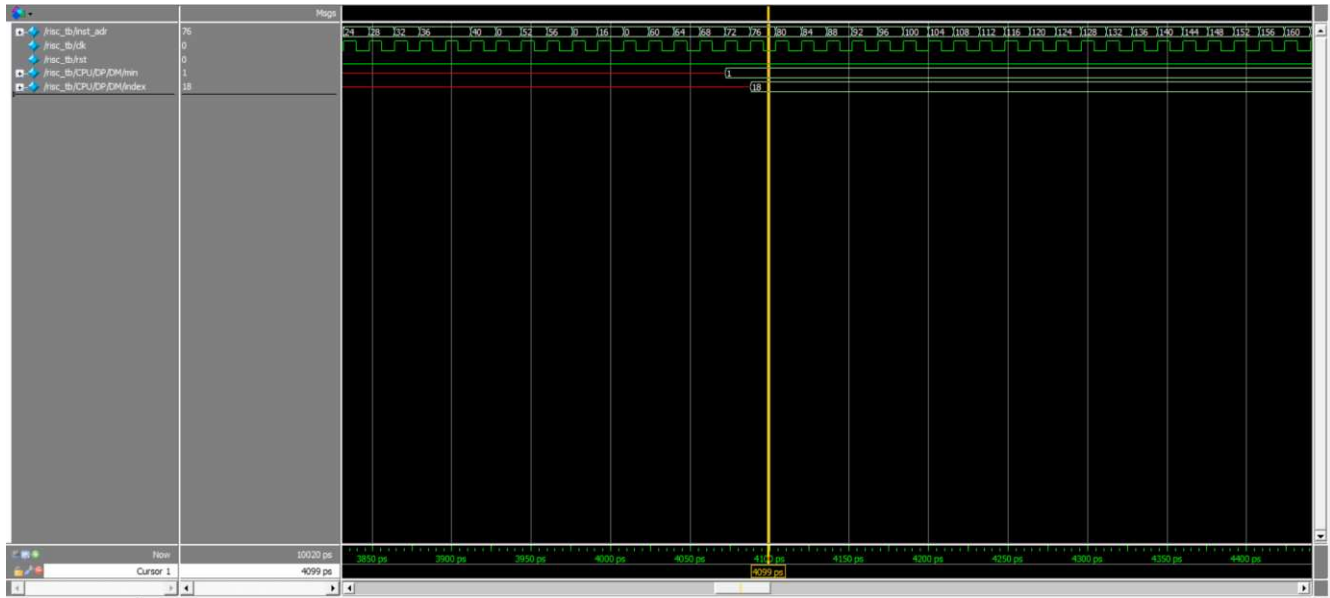
Min = -111

Index = 1

```
64'd8;      //0
-64'd111;   //1
64'd11;     //2
64'd5;      //3
64'd4;      //4
64'd5;      //5
64'd6;      //6
64'd7;      //7
64'd7;      //8
64'd3;      //9
64'd101;    //10
64'd11;     //11
64'd12;     //12
64'd13;     //13
64'd5;      //14
64'd14;     //15
64'd14;     //16
64'd11;     //17
64'd11;     //18
64'd11;     //19
```

Min = 1

Index = 18

```
64'd8;      //0
64'd111;    //1
64'd11;     //2
64'd5;      //3
64'd4;      //4
64'd5;      //5
64'd6;      //6
64'd7;      //7
64'd7;      //8
64'd3;      //9
64'd101;    //10
64'd11;     //11
64'd12;     //12
64'd13;     //13
64'd5;      //14
64'd14;     //15
64'd14;     //16
64'd11;     //17
64'd1;      //18
64'd11;     //19
```
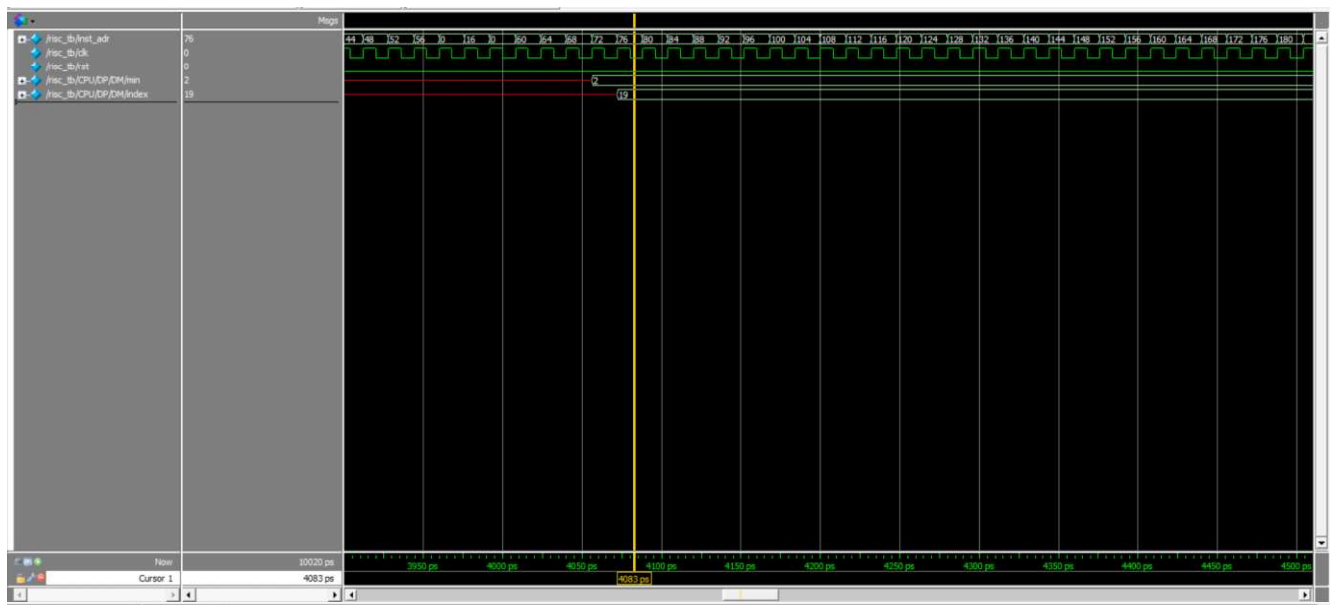
Min = 2

Index = 19

```
64'd8;      //0
64'd111;    //1
64'd11;     //2
64'd5;      //3
64'd4;      //4
64'd5;      //5
64'd6;      //6
64'd7;      //7
64'd7;      //8
64'd13;     //9
64'd101;    //10
64'd11;     //11
64'd12;     //12
64'd13;     //13
64'd5;      //14
64'd14;     //15
64'd14;     //16
64'd11;     //17
64'd11;     //18
64'd2; //19
```

Min = -7

Index = 7

```
64'd8;       //0
64'd111;     //1
64'd11;      //2
64'd5;       //3
64'd14;      //4
64'd15;      //5
64'd6;       //6
-64'd7;      //7
64'd5;       //8
64'd13;      //9
64'd101;     //10
64'd11;      //11
64'd12;      //12
64'd13;      //13
64'd115;     //14
64'd14;      //15
64'd14;      //16
64'd11;      //17
64'd11;      //18
64'd10;      //19
```
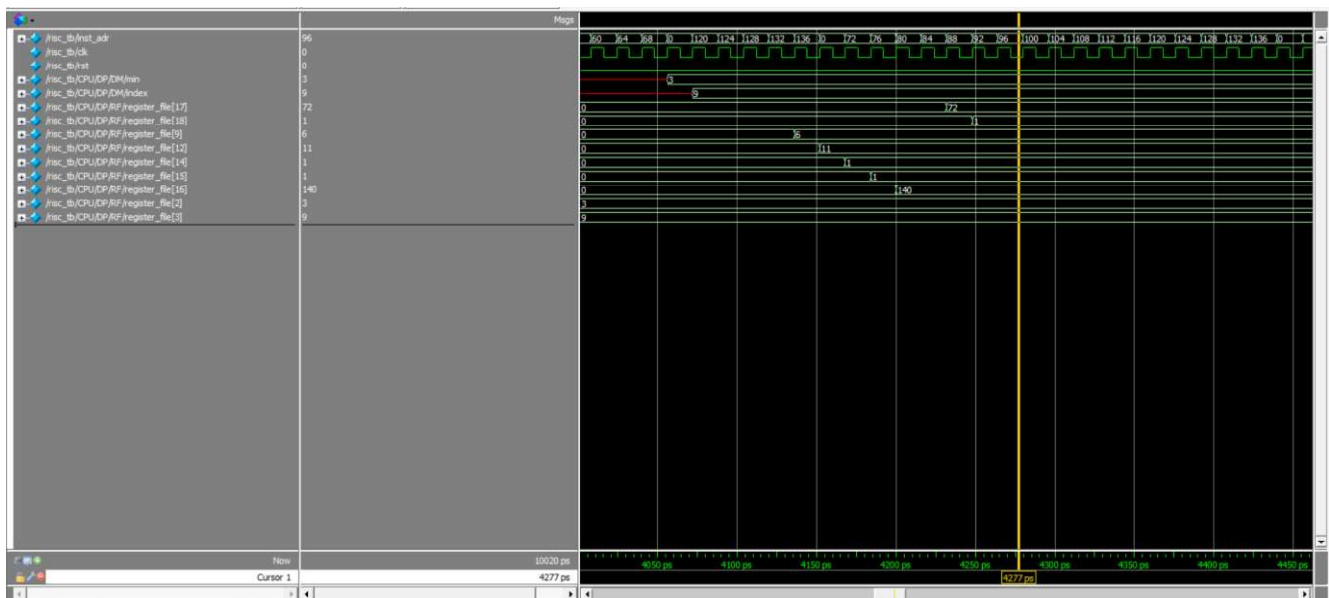
In order to check other instructions such as sll and slr and jalr and jal and … we used this code:

```
//              jal  X1, TEST        68
//   RETURTN:   sll R17,R3,R2        72
//              srl R18,R3,R2        76

//   TEST:      sub R9,R3,R2         120
//              or R12,R3,R2         124
//              and R14,R3,R2        128
//              slti R15,R2,20       132
//              jalr R16 RETURN(R2)  136
```


The results are totally correct.