计算机体系结构实验

MIPS PIPELINE

流水线 cpu 设计实验

测

试

手

册

实验组成员：张瑞祎，徐栋，李乾科

手册编者：李乾科

2014-6-10

目　　录

一、　　　　单条指令测试

State：

单条指令测试分为四个部分：
R-type 指令测试、I-type 指令测试、sw 与 lw 及其相关指令测试、跳转及分支指令测试

其中 R-type 指令与 I-type 指令共 15 种指令，在一个测试文件中测试，测试文件共 18 条指令，其中多出的三条为 add，addi，sub 溢出的情况测试。在这 15 张截图中，为了方便观察验证，每条指令只截取了从取指令开始的四个周期。图中显示包括：IR，ALU、SHIFT 的输入输出，ID 段中有关寄存器的输入输出，Data 输出，跳转控制信号，PC 的输入输出等信号。

sw 与 lw 及其相关指令测试中我们测试了 sw,swl,swr,lw,lwl,lwr 六种指令。为了更好的显示测试，我们显示了更多的信号，包括：IR，ALU、SHIFT 的输入输出，ID 段中有关寄存器的输入输出，Data 输出，跳转控制信号，PC 的输入输出，数据内存相关的寄存器值，立即数的值以及是否冲刷的控制信号等。

跳转及分支指令测试，共有 begz,begzal,beq,j 四条指令，信号输出同 sw、lw 指令，其他测试说明在测试图后会说明

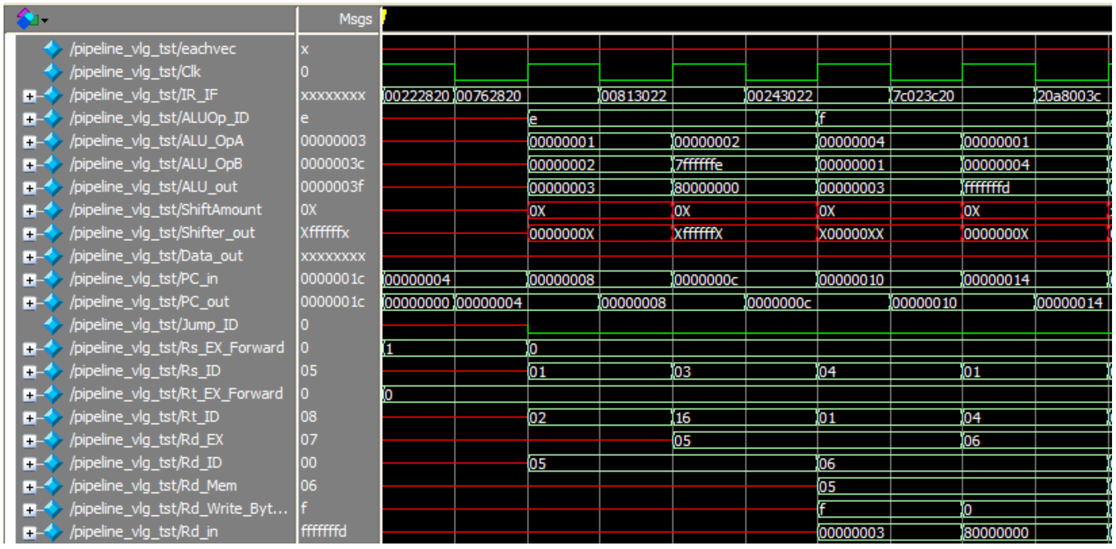
1、R-type 指令

initial

r1 = 1 ; r2 = 2 ; r3 = 3 ; r20 = 0x7ffffffc ; r21 = 0x7ffffffd ; r22 = 0x7ffffffe ;

add

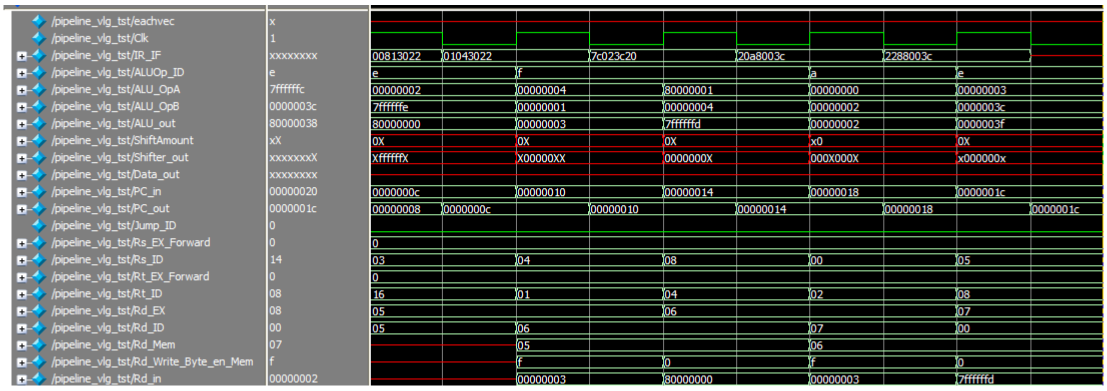
ram[0]=32'b00000000001000100010100000100000;　//add　r1 r2 = 3 -> r5　不溢出
ram[1]=32'b000000_00011_10110_0010100000100000;　//add　r3 r22 -> r5　溢出


Msgs
/pipeline_vlg_tst/eachvec x
/pipeline_vlg_tst/Clk 0
/pipeline_vlg_tst/IR_IF xxxxxxxx
/pipeline_vlg_tst/ALUOp_ID e
/pipeline_vlg_tst/ALU_OpA 00000003
/pipeline_vlg_tst/ALU_OpB 0000003c
/pipeline_vlg_tst/ALU_out 0000003f
/pipeline_vlg_tst/ShiftAmount 0X
/pipeline_vlg_tst/Shifter_out Xffffffx
/pipeline_vlg_tst/Data_out xxxxxxxx
/pipeline_vlg_tst/PC_in 0000001c
/pipeline_vlg_tst/PC_out 0000001c
/pipeline_vlg_tst/Jump_ID 0
/pipeline_vlg_tst/Rs_EX_Forward 0
/pipeline_vlg_tst/Rs_ID 05
/pipeline_vlg_tst/Rt_EX_Forward 0
/pipeline_vlg_tst/Rt_ID 08
/pipeline_vlg_tst/Rd_EX 07
/pipeline_vlg_tst/Rd_ID 00
/pipeline_vlg_tst/Rd_Mem 06
/pipeline_vlg_tst/Rd_Write_Byt... f
/pipeline_vlg_tst/Rd_in fffffffd
00222820 00762820 00813022 00243022 7c023c20 20a8003c
e f
00000001 00000002 00000004 00000001
00000002 7ffffffe 00000001 00000004
00000003 80000000 00000003 fffffffd
0X 0X 0X 0X
0000000X XffffffX X00000XX 0000000X
00000004 00000008 0000000c 00000010 00000014
00000000 00000004 00000008 0000000c 00000010 00000014
1 0
01 03 04 01
0
02 16 01 04
05 06
05 06
05
f 0
00000003 80000000

sub

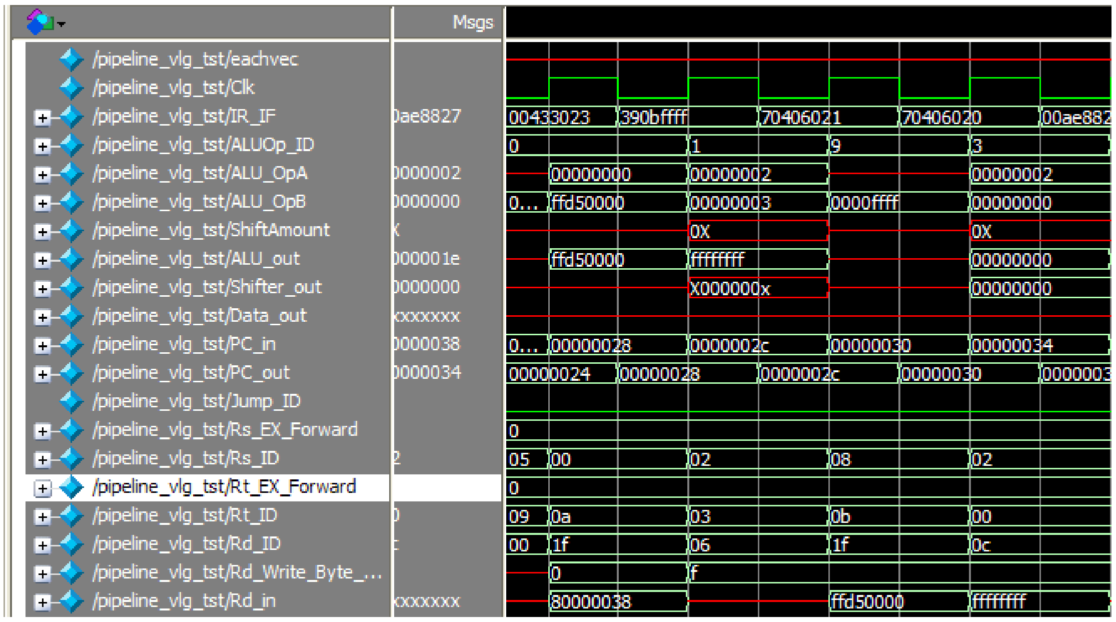
ram[2]=32'b00000000010000001001100000100010;　//sub　r4 r1 = 3 -> r6　不溢出

ram[3]=32'b000000_00001_00100_0011000000100010;　//sub　r1 r4 = -3 -> r6　溢出


/pipeline_vlg_tst/eachvec
/pipeline_vlg_tst/Clk
/pipeline_vlg_tst/IR_IF
/pipeline_vlg_tst/ALUOp_ID
/pipeline_vlg_tst/ALU_OpA
/pipeline_vlg_tst/ALU_OpB
/pipeline_vlg_tst/ALU_out
/pipeline_vlg_tst/ShiftAmount
/pipeline_vlg_tst/Shifter_out
/pipeline_vlg_tst/Data_out
/pipeline_vlg_tst/PC_in
/pipeline_vlg_tst/PC_out
/pipeline_vlg_tst/Jump_ID
/pipeline_vlg_tst/Rs_EX_Forward
/pipeline_vlg_tst/Rs_ID
/pipeline_vlg_tst/Rt_EX_Forward
/pipeline_vlg_tst/Rt_ID
/pipeline_vlg_tst/Rd_EX
/pipeline_vlg_tst/Rd_ID
/pipeline_vlg_tst/Rd_Mem
/pipeline_vlg_tst/Rd_Write_Byte_en_Mem
/pipeline_vlg_tst/Rd_in

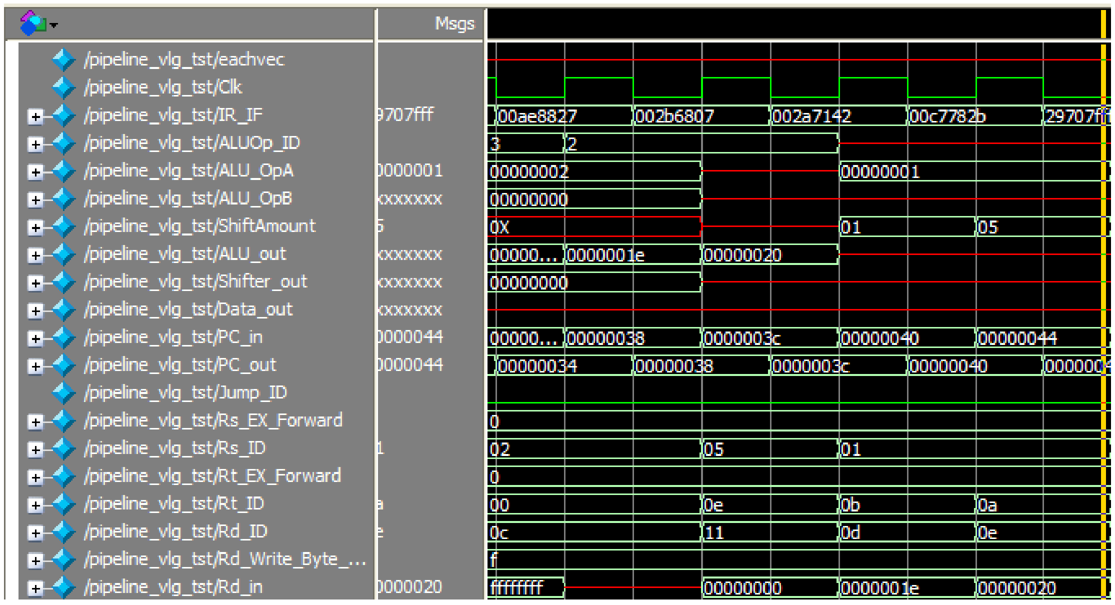
subu

ram[9]=32'b00000000010000110011000000100011;　//subu r2 r3 = -1 -> r6


Msgs
/pipeline_vlg_tst/eachvec
/pipeline_vlg_tst/Clk
/pipeline_vlg_tst/IR_IF
/pipeline_vlg_tst/ALUOp_ID
/pipeline_vlg_tst/ALU_OpA
/pipeline_vlg_tst/ALU_OpB
/pipeline_vlg_tst/ShiftAmount
/pipeline_vlg_tst/ALU_out
/pipeline_vlg_tst/Shifter_out
/pipeline_vlg_tst/Data_out
/pipeline_vlg_tst/PC_in
/pipeline_vlg_tst/PC_out
/pipeline_vlg_tst/Jump_ID
/pipeline_vlg_tst/Rs_EX_Forward
/pipeline_vlg_tst/Rs_ID
/pipeline_vlg_tst/Rt_EX_Forward
/pipeline_vlg_tst/Rt_ID
/pipeline_vlg_tst/Rd_ID
/pipeline_vlg_tst/Rd_Write_Byte_...
/pipeline_vlg_tst/Rd_in

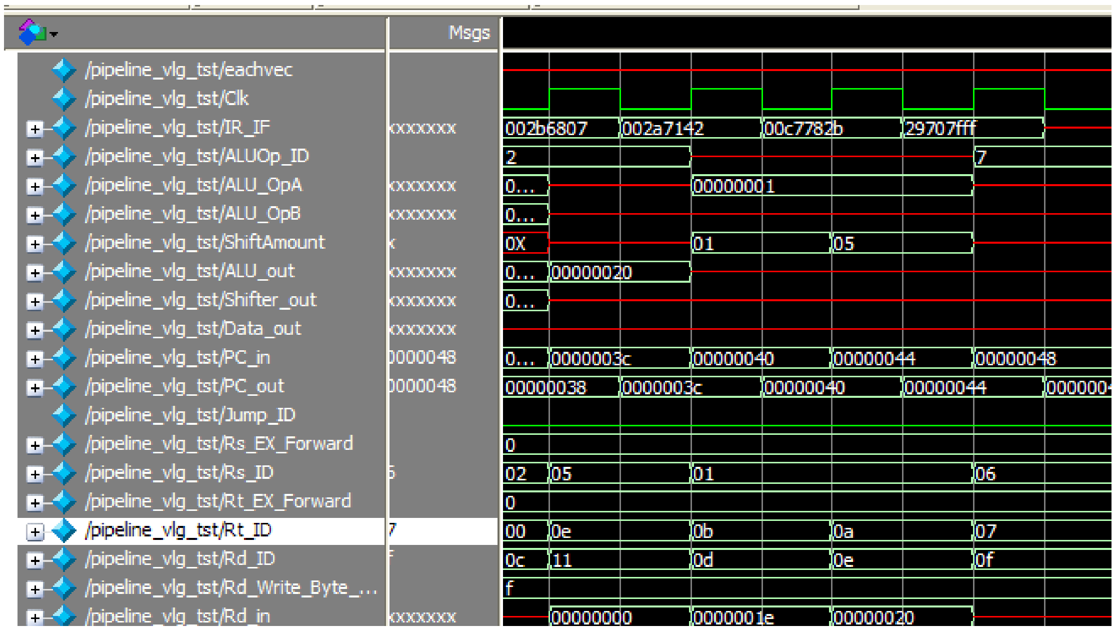
nor

ram[13]=32'b00000000101011101000100000100111; //nor r5 r14 -> r17


Msgs
/pipeline_vlg_tst/eachvec
/pipeline_vlg_tst/Clk
/pipeline_vlg_tst/IR_IF
/pipeline_vlg_tst/ALUOp_ID
/pipeline_vlg_tst/ALU_OpA
/pipeline_vlg_tst/ALU_OpB
/pipeline_vlg_tst/ShiftAmount
/pipeline_vlg_tst/ALU_out
/pipeline_vlg_tst/Shifter_out
/pipeline_vlg_tst/Data_out
/pipeline_vlg_tst/PC_in
/pipeline_vlg_tst/PC_out
/pipeline_vlg_tst/Jump_ID
/pipeline_vlg_tst/Rs_EX_Forward
/pipeline_vlg_tst/Rs_ID
/pipeline_vlg_tst/Rt_EX_Forward
/pipeline_vlg_tst/Rt_ID
/pipeline_vlg_tst/Rd_ID
/pipeline_vlg_tst/Rd_Write_Byte_...
/pipeline_vlg_tst/Rd_in
00ae8827 002b6807 002a7142 00c7782b 29707fff
00000002 00000001
00000000
0X 01 05
0000001e 00000020
00000038 0000003c 00000040 00000044
00000034 00000038 0000003c 00000040
02 05 01
00 0e 0b 0a
0c 11 0d 0e
ffffffff 00000000 0000001e 00000020

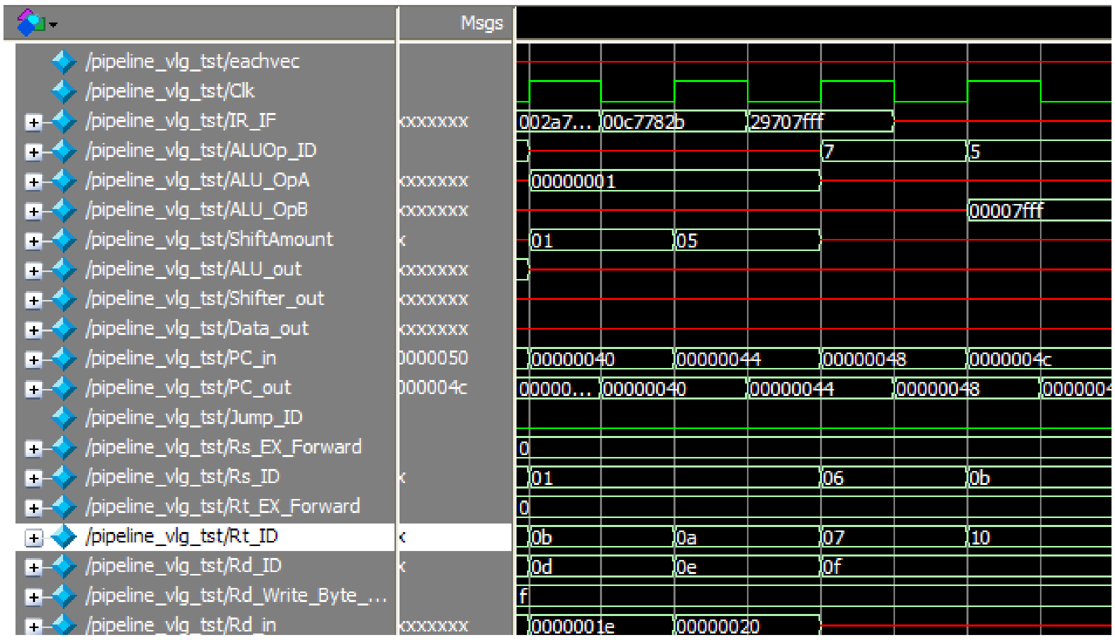
srav

ram[14]=32'b00000000001010110110100000000111; //srav r1 r11 -> r13


Msgs
/pipeline_vlg_tst/eachvec
/pipeline_vlg_tst/Clk
/pipeline_vlg_tst/IR_IF
/pipeline_vlg_tst/ALUOp_ID
/pipeline_vlg_tst/ALU_OpA
/pipeline_vlg_tst/ALU_OpB
/pipeline_vlg_tst/ShiftAmount
/pipeline_vlg_tst/ALU_out
/pipeline_vlg_tst/Shifter_out
/pipeline_vlg_tst/Data_out
/pipeline_vlg_tst/PC_in
/pipeline_vlg_tst/PC_out
/pipeline_vlg_tst/Jump_ID
/pipeline_vlg_tst/Rs_EX_Forward
/pipeline_vlg_tst/Rs_ID
/pipeline_vlg_tst/Rt_EX_Forward
/pipeline_vlg_tst/Rt_ID
/pipeline_vlg_tst/Rd_ID
/pipeline_vlg_tst/Rd_Write_Byte_...
/pipeline_vlg_tst/Rd_in
002b6807 002a7142 00c7782b 29707fff
2 7
00000001
0X 01 05
00000020
0000003c 00000040 00000044 00000048
00000038 0000003c 00000040 00000044
02 05 01 06
00 0e 0b 0a 07
0c 11 0d 0e 0f
00000000 0000001e 00000020

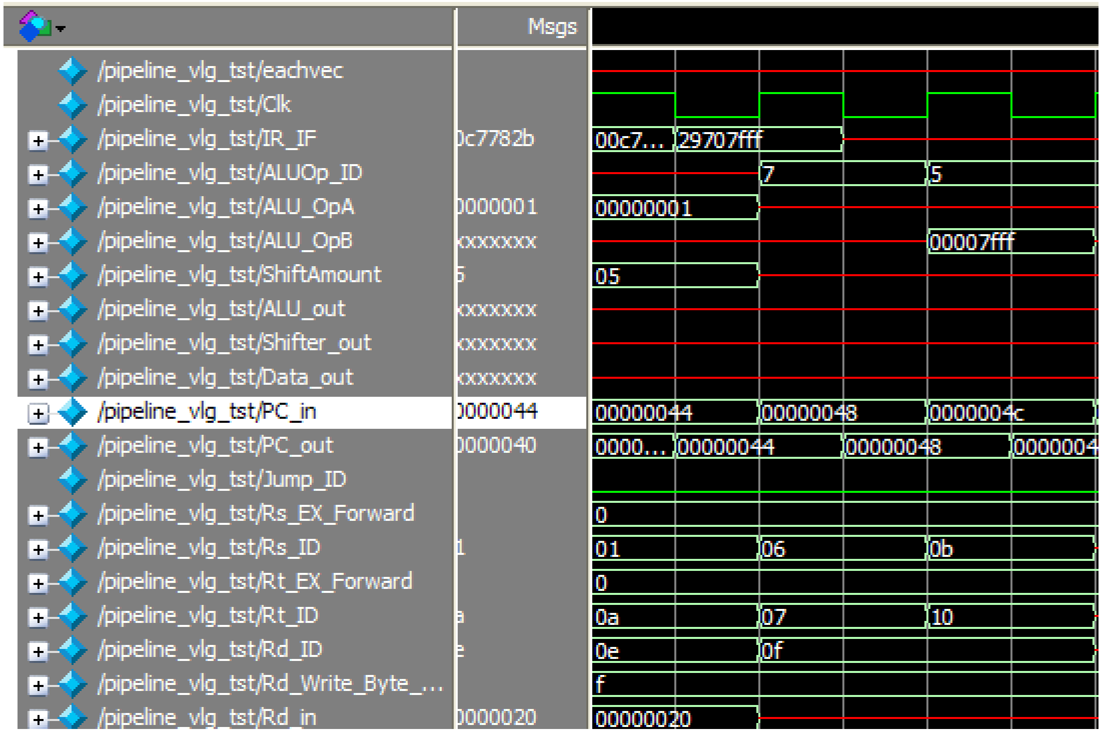
rotr

ram[15]=32'b00000000001010100111000101000010; //rotr r10 5 -> r14


Msgs
/pipeline_vlg_tst/eachvec
/pipeline_vlg_tst/Clk
/pipeline_vlg_tst/IR_IF
/pipeline_vlg_tst/ALUOp_ID
/pipeline_vlg_tst/ALU_OpA
/pipeline_vlg_tst/ALU_OpB
/pipeline_vlg_tst/ShiftAmount
/pipeline_vlg_tst/ALU_out
/pipeline_vlg_tst/Shifter_out
/pipeline_vlg_tst/Data_out
/pipeline_vlg_tst/PC_in
/pipeline_vlg_tst/PC_out
/pipeline_vlg_tst/Jump_ID
/pipeline_vlg_tst/Rs_EX_Forward
/pipeline_vlg_tst/Rs_ID
/pipeline_vlg_tst/Rt_EX_Forward
/pipeline_vlg_tst/Rt_ID
/pipeline_vlg_tst/Rd_ID
/pipeline_vlg_tst/Rd_Write_Byte_...
/pipeline_vlg_tst/Rd_in
002a7... 00c7782b 29707fff
7 5
00000001
00007fff
01 05
00000040 00000044 00000048 0000004c
00000... 00000040 00000044 00000048
0
01 06 0b
0
0b 0a 07 10
0d 0e 0f
f
0000001e 00000020

sltu

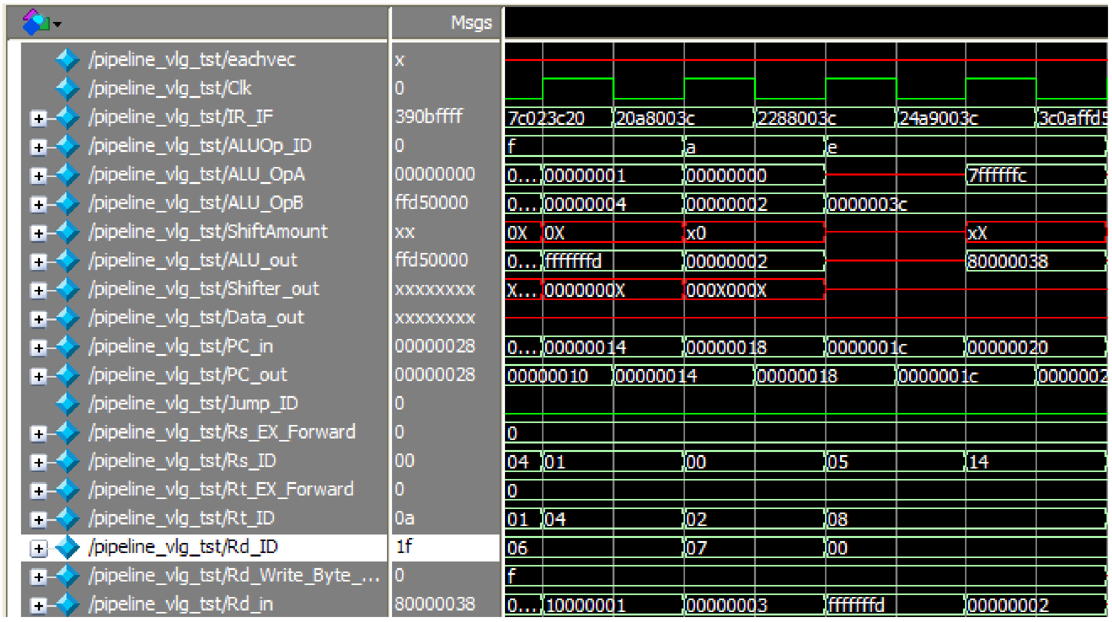
ram[16]=32'b00000000110001110111100000101011; //sltu r12 r13 -> r15


Msgs
/pipeline_vlg_tst/eachvec
/pipeline_vlg_tst/Clk
/pipeline_vlg_tst/IR_IF
/pipeline_vlg_tst/ALUOp_ID
/pipeline_vlg_tst/ALU_OpA
/pipeline_vlg_tst/ALU_OpB
/pipeline_vlg_tst/ShiftAmount
/pipeline_vlg_tst/ALU_out
/pipeline_vlg_tst/Shifter_out
/pipeline_vlg_tst/Data_out
/pipeline_vlg_tst/PC_in
/pipeline_vlg_tst/PC_out
/pipeline_vlg_tst/Jump_ID
/pipeline_vlg_tst/Rs_EX_Forward
/pipeline_vlg_tst/Rs_ID
/pipeline_vlg_tst/Rt_EX_Forward
/pipeline_vlg_tst/Rt_ID
/pipeline_vlg_tst/Rd_ID
/pipeline_vlg_tst/Rd_Write_Byte_...
/pipeline_vlg_tst/Rd_in
00c7... 29707fff
7 5
00000001
00007fff
05
00000044 00000048 0000004c
0000... 00000044 00000048
0
01 06 0b
0
0a 07 10
0e 0f
f
00000020

2、I-type 指令

seb

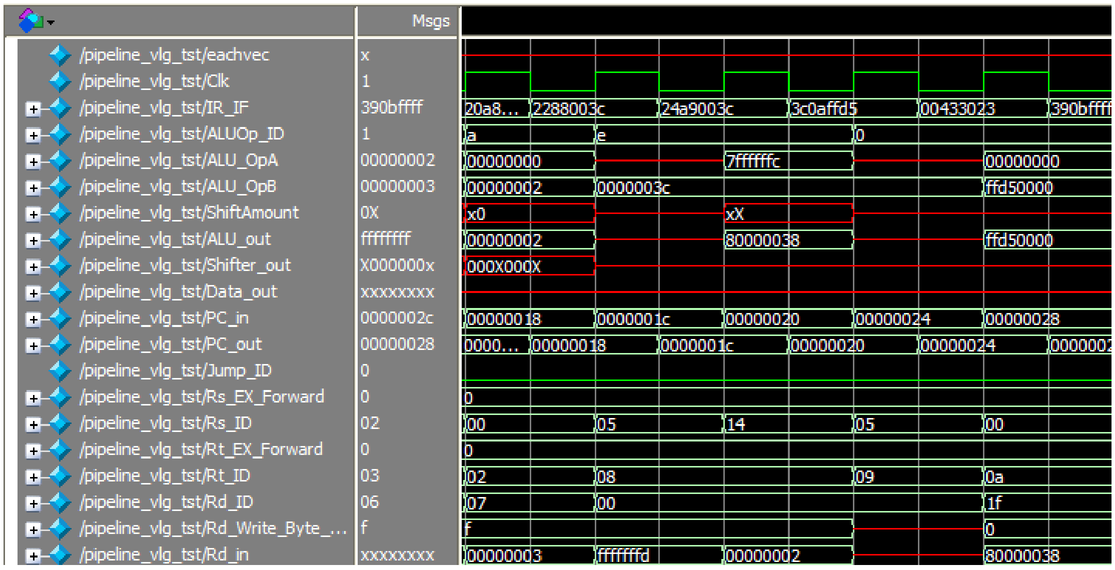
ram[4]=32'b01111100000000010001110000100000; //seb r2   = 0 -> r7


Msgs
/pipeline_vlg_tst/eachvec x
/pipeline_vlg_tst/Clk 0
/pipeline_vlg_tst/IR_IF 390bffff
/pipeline_vlg_tst/ALUOp_ID 0
/pipeline_vlg_tst/ALU_OpA 00000000
/pipeline_vlg_tst/ALU_OpB ffd50000
/pipeline_vlg_tst/ShiftAmount xx
/pipeline_vlg_tst/ALU_out ffd50000
/pipeline_vlg_tst/Shifter_out xxxxxxxx
/pipeline_vlg_tst/Data_out xxxxxxxx
/pipeline_vlg_tst/PC_in 00000028
/pipeline_vlg_tst/PC_out 00000028
/pipeline_vlg_tst/Jump_ID 0
/pipeline_vlg_tst/Rs_EX_Forward 0
/pipeline_vlg_tst/Rs_ID 00
/pipeline_vlg_tst/Rt_EX_Forward 0
/pipeline_vlg_tst/Rt_ID 0a
/pipeline_vlg_tst/Rd_ID 1f
/pipeline_vlg_tst/Rd_Write_Byte_... 0
/pipeline_vlg_tst/Rd_in 80000038

addi

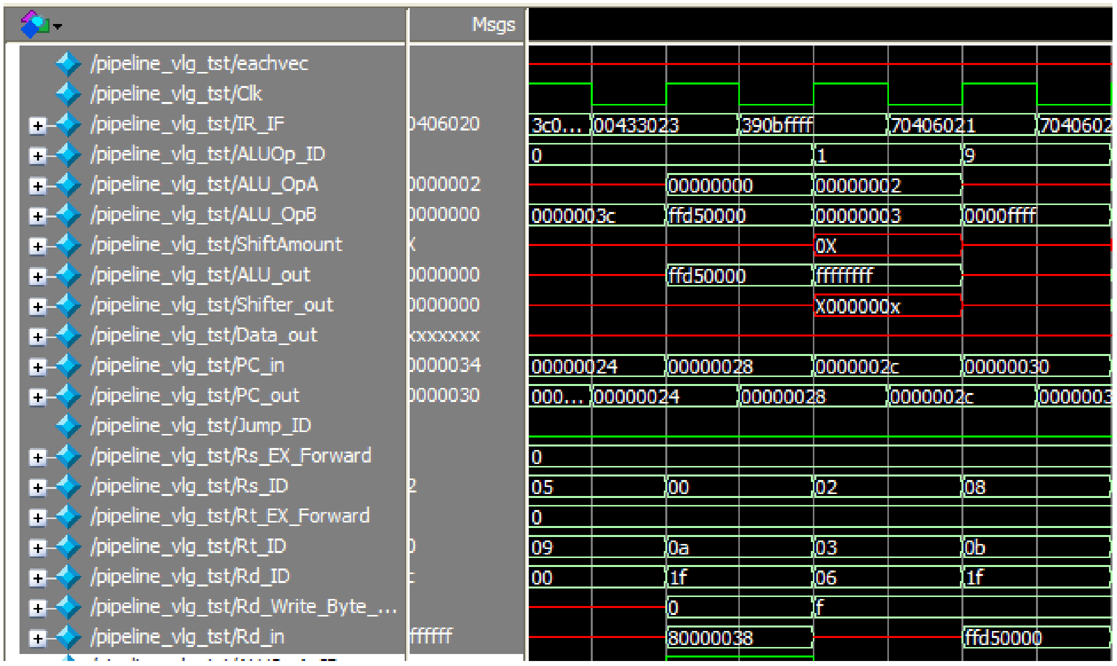
ram[5]=32'b00100000101010000000000000111100; //addi r5 im = 111111 -> r8 溢出

ram[6]=32'b001000_10100_01000_0000000000111100; //addi r20 im = 111111 -> r8 不溢出


Msgs
/pipeline_vlg_tst/eachvec x
/pipeline_vlg_tst/Clk 1
/pipeline_vlg_tst/IR_IF 390bffff
/pipeline_vlg_tst/ALUOp_ID 1
/pipeline_vlg_tst/ALU_OpA 00000002
/pipeline_vlg_tst/ALU_OpB 00000003
/pipeline_vlg_tst/ShiftAmount 0X
/pipeline_vlg_tst/ALU_out ffffffff
/pipeline_vlg_tst/Shifter_out X000000x
/pipeline_vlg_tst/Data_out xxxxxxxx
/pipeline_vlg_tst/PC_in 0000002c
/pipeline_vlg_tst/PC_out 00000028
/pipeline_vlg_tst/Jump_ID 0
/pipeline_vlg_tst/Rs_EX_Forward 0
/pipeline_vlg_tst/Rs_ID 02
/pipeline_vlg_tst/Rt_EX_Forward 0
/pipeline_vlg_tst/Rt_ID 03
/pipeline_vlg_tst/Rd_ID 06
/pipeline_vlg_tst/Rd_Write_Byte_... f
/pipeline_vlg_tst/Rd_in xxxxxxxx

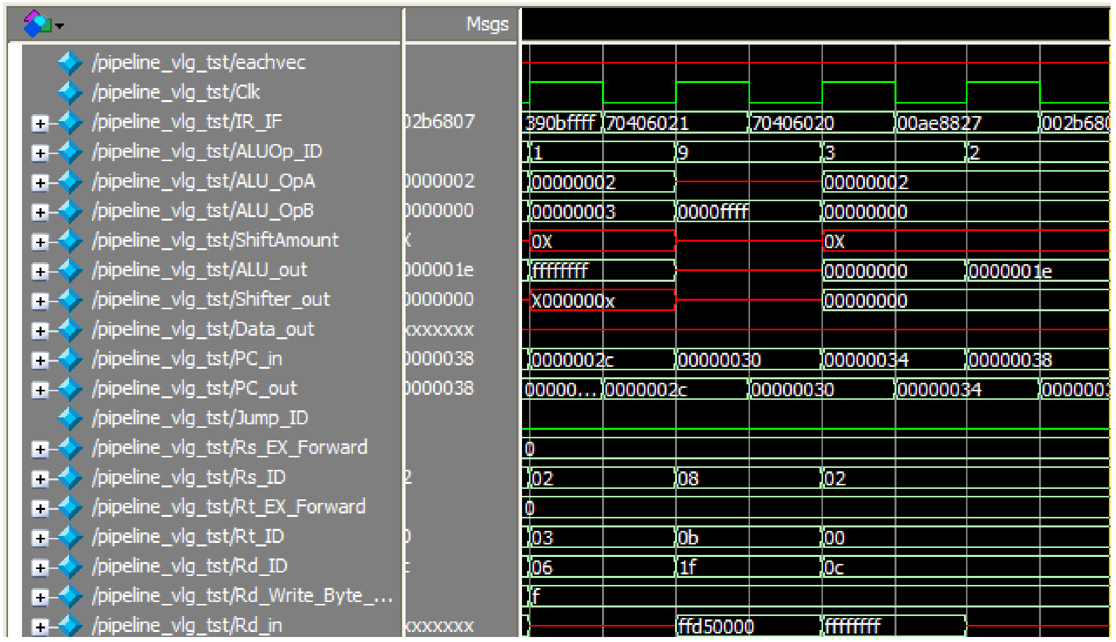
lui

ram[8]=32'b00111100000010101111111111010101; //lui 1111111111010101 -> r10


Msgs
/pipeline_vlg_tst/eachvec
/pipeline_vlg_tst/Clk
/pipeline_vlg_tst/IR_IF
/pipeline_vlg_tst/ALUOp_ID
/pipeline_vlg_tst/ALU_OpA
/pipeline_vlg_tst/ALU_OpB
/pipeline_vlg_tst/ShiftAmount
/pipeline_vlg_tst/ALU_out
/pipeline_vlg_tst/Shifter_out
/pipeline_vlg_tst/Data_out
/pipeline_vlg_tst/PC_in
/pipeline_vlg_tst/PC_out
/pipeline_vlg_tst/Jump_ID
/pipeline_vlg_tst/Rs_EX_Forward
/pipeline_vlg_tst/Rs_ID
/pipeline_vlg_tst/Rt_EX_Forward
/pipeline_vlg_tst/Rt_ID
/pipeline_vlg_tst/Rd_ID
/pipeline_vlg_tst/Rd_Write_Byte_…
/pipeline_vlg_tst/Rd_in

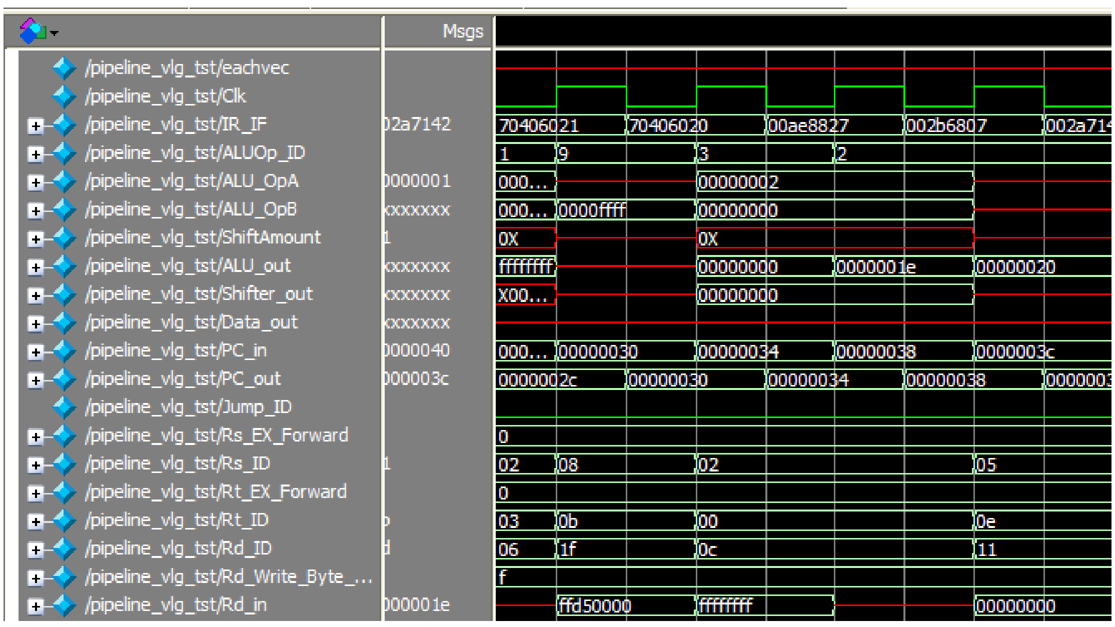
xori

ram[10]=32'b001110010000101111111111111111111;   //xori 1111111110111110 -> r11


Msgs
/pipeline_vlg_tst/eachvec
/pipeline_vlg_tst/Clk
/pipeline_vlg_tst/IR_IF
/pipeline_vlg_tst/ALUOp_ID
/pipeline_vlg_tst/ALU_OpA
/pipeline_vlg_tst/ALU_OpB
/pipeline_vlg_tst/ShiftAmount
/pipeline_vlg_tst/ALU_out
/pipeline_vlg_tst/Shifter_out
/pipeline_vlg_tst/Data_out
/pipeline_vlg_tst/PC_in
/pipeline_vlg_tst/PC_out
/pipeline_vlg_tst/Jump_ID
/pipeline_vlg_tst/Rs_EX_Forward
/pipeline_vlg_tst/Rs_ID
/pipeline_vlg_tst/Rt_EX_Forward
/pipeline_vlg_tst/Rt_ID
/pipeline_vlg_tst/Rd_ID
/pipeline_vlg_tst/Rd_Write_Byte_…
/pipeline_vlg_tst/Rd_in

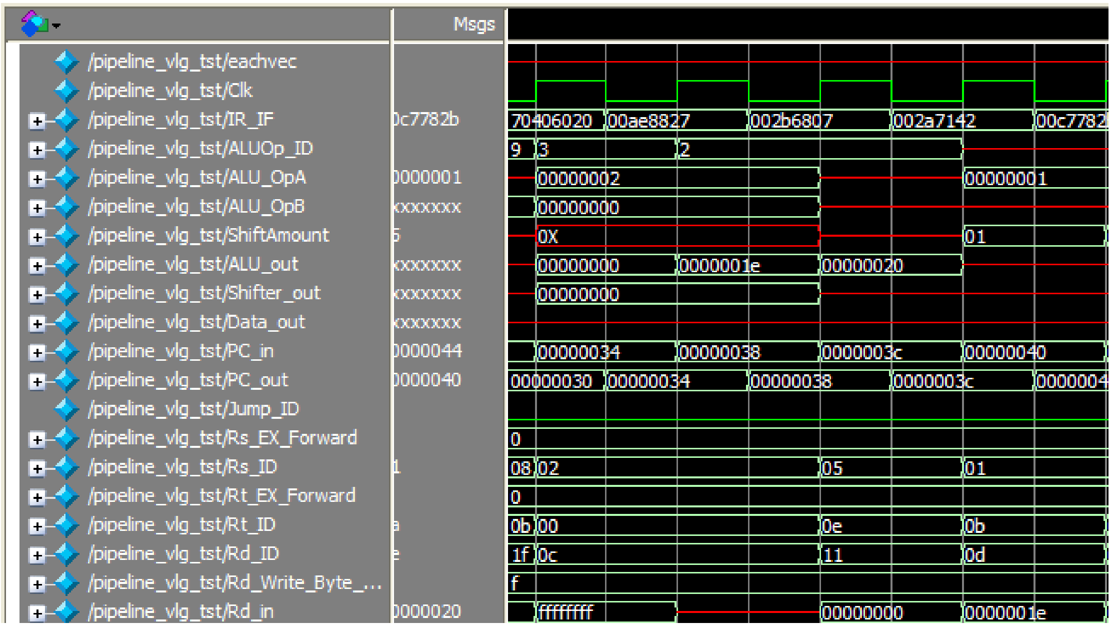
clo

ram[11]=32'b011100_00010_00000_0110000000100001;   //clo r2 = 0 -> r12


Msgs
/pipeline_vlg_tst/eachvec
/pipeline_vlg_tst/Clk
/pipeline_vlg_tst/IR_IF
/pipeline_vlg_tst/ALUOp_ID
/pipeline_vlg_tst/ALU_OpA
/pipeline_vlg_tst/ALU_OpB
/pipeline_vlg_tst/ShiftAmount
/pipeline_vlg_tst/ALU_out
/pipeline_vlg_tst/Shifter_out
/pipeline_vlg_tst/Data_out
/pipeline_vlg_tst/PC_in
/pipeline_vlg_tst/PC_out
/pipeline_vlg_tst/Jump_ID
/pipeline_vlg_tst/Rs_EX_Forward
/pipeline_vlg_tst/Rs_ID
/pipeline_vlg_tst/Rt_EX_Forward
/pipeline_vlg_tst/Rt_ID
/pipeline_vlg_tst/Rd_ID
/pipeline_vlg_tst/Rd_Write_Byte_…
/pipeline_vlg_tst/Rd_in

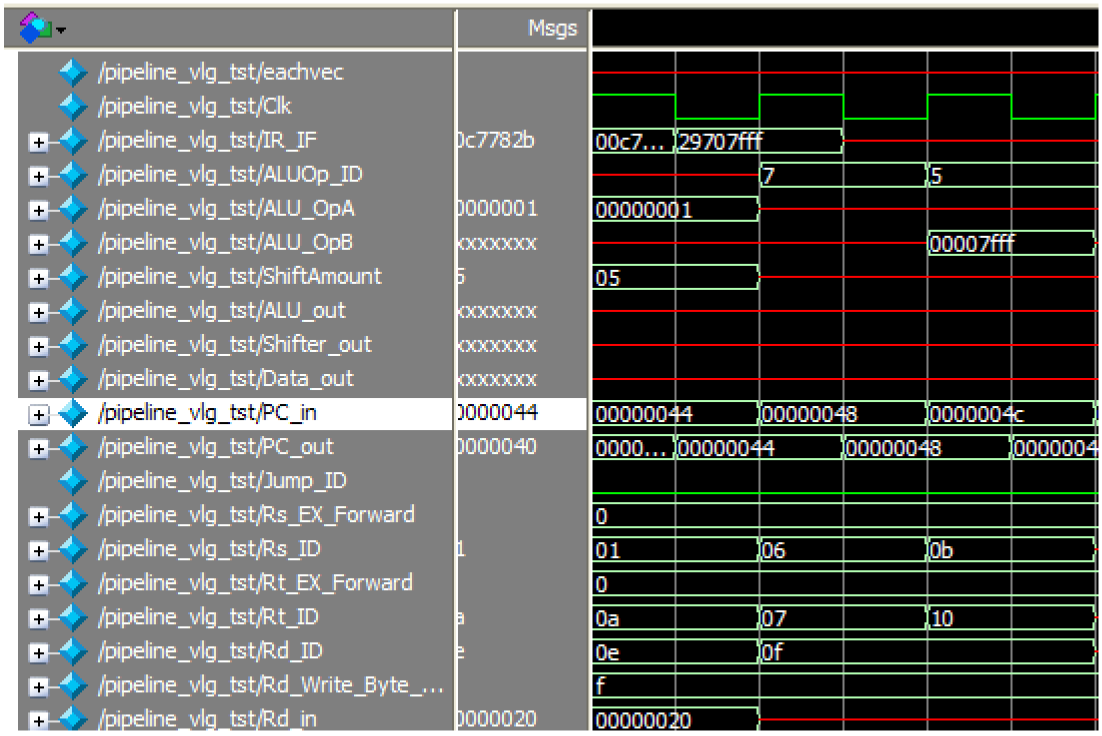
clz

ram[12]=32'b011100_00010_00000_01100_00000100000;   //clz r2 = 30 -> r12


Msgs
/pipeline_vlg_tst/eachvec
/pipeline_vlg_tst/Clk
/pipeline_vlg_tst/IR_IF
/pipeline_vlg_tst/ALUOp_ID
/pipeline_vlg_tst/ALU_OpA
/pipeline_vlg_tst/ALU_OpB
/pipeline_vlg_tst/ShiftAmount
/pipeline_vlg_tst/ALU_out
/pipeline_vlg_tst/Shifter_out
/pipeline_vlg_tst/Data_out
/pipeline_vlg_tst/PC_in
/pipeline_vlg_tst/PC_out
/pipeline_vlg_tst/Jump_ID
/pipeline_vlg_tst/Rs_EX_Forward
/pipeline_vlg_tst/Rs_ID
/pipeline_vlg_tst/Rt_EX_Forward
/pipeline_vlg_tst/Rt_ID
/pipeline_vlg_tst/Rd_ID
/pipeline_vlg_tst/Rd_Write_Byte_…
/pipeline_vlg_tst/Rd_in

slti

ram[17]=32'b00101001011100000111111111111111; //slti r11 0x7fff -> r16



```
                                              Msgs
/pipeline_vlg_tst/eachvec
/pipeline_vlg_tst/Clk
/pipeline_vlg_tst/IR_IF                      0c7782b     00c7...  29707fff
/pipeline_vlg_tst/ALUOp_ID                                         7         5
/pipeline_vlg_tst/ALU_OpA                    0000001     00000001
/pipeline_vlg_tst/ALU_OpB                    xxxxxxx                         00007fff
/pipeline_vlg_tst/ShiftAmount                5           05
/pipeline_vlg_tst/ALU_out                    xxxxxxx
/pipeline_vlg_tst/Shifter_out                xxxxxxx
/pipeline_vlg_tst/Data_out                   xxxxxxx
/pipeline_vlg_tst/PC_in                      0000044     00000044  00000048  0000004c
/pipeline_vlg_tst/PC_out                     0000040     0000...  00000044  00000048  0000004
/pipeline_vlg_tst/Jump_ID
/pipeline_vlg_tst/Rs_EX_Forward                          0
/pipeline_vlg_tst/Rs_ID                      1           01        06        0b
/pipeline_vlg_tst/Rt_EX_Forward                          0
/pipeline_vlg_tst/Rt_ID                      a           0a        07        10
/pipeline_vlg_tst/Rd_ID                      e           0e        0f
/pipeline_vlg_tst/Rd_Write_Byte_...                      f
/pipeline_vlg_tst/Rd_in                      0000020     00000020
```

3、sw 、 lw 及其相关指令测试

initial

register[0] = 0;

register[1] = 1;

register[2] = 5;

register[3] = 4;

register[4] = 3;

register[5] = 2;

register[8] = 32'hfffffffe;

register[9] = 9;

register[20] = 32'h7ffffffc;

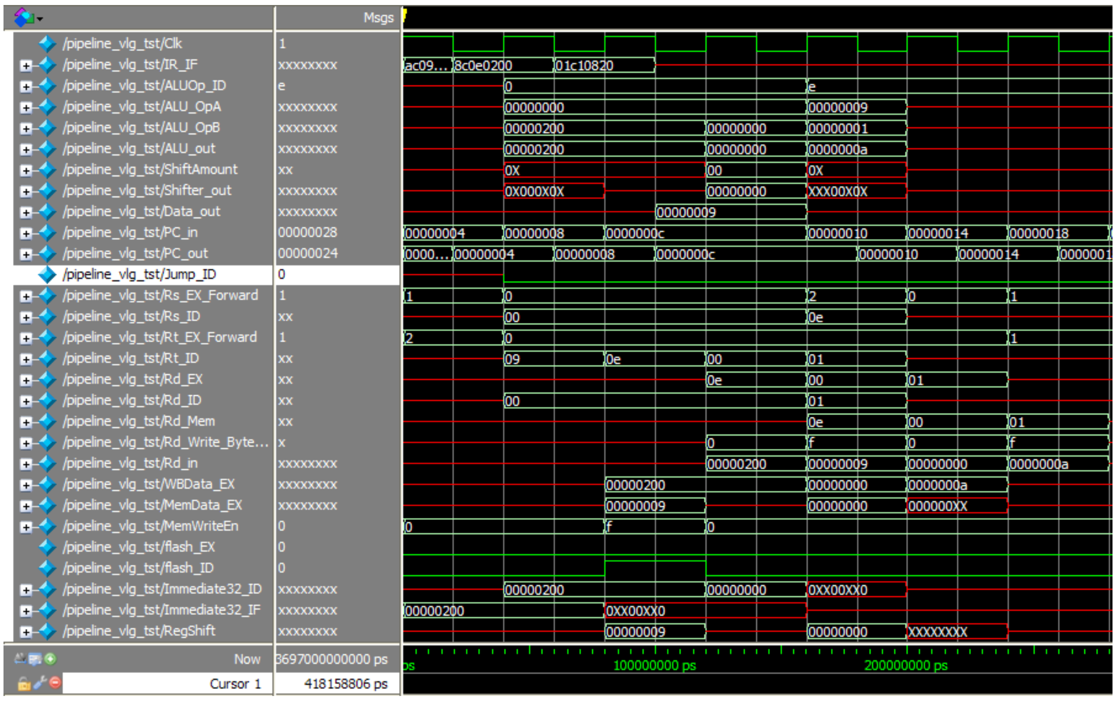
register[21] = 32'h7ffffffd;

register[22] = 32'hfffffffe;

ram[0] = {16'b101011_00000_01001,16'd512};//sw reg[9] -> IM[512]

ram[1] = {16'b100011_00000_01110,16'd512};//lw IM[512] -> reg[14]

ram[2] = 32'b000000_01110_00001_00001_00000_100000; add r14 + r1 ->r1;



```
/pipeline_vlg_tst/Clk                  1
/pipeline_vlg_tst/IR_IF                xxxxxxxx
/pipeline_vlg_tst/ALUOp_ID             e
/pipeline_vlg_tst/ALU_OpA              xxxxxxxx
/pipeline_vlg_tst/ALU_OpB              xxxxxxxx
/pipeline_vlg_tst/ALU_out              xxxxxxxx
/pipeline_vlg_tst/ShiftAmount          xx
/pipeline_vlg_tst/Shifter_out          xxxxxxxx
/pipeline_vlg_tst/Data_out             xxxxxxxx
/pipeline_vlg_tst/PC_in                00000028
/pipeline_vlg_tst/PC_out               00000024
/pipeline_vlg_tst/Jump_ID              0
/pipeline_vlg_tst/Rs_EX_Forward        1
/pipeline_vlg_tst/Rs_ID                xx
/pipeline_vlg_tst/Rt_EX_Forward        1
/pipeline_vlg_tst/Rt_ID                xx
/pipeline_vlg_tst/Rd_EX                xx
/pipeline_vlg_tst/Rd_ID                xx
/pipeline_vlg_tst/Rd_Mem               xx
/pipeline_vlg_tst/Rd_Write_Byte...     x
/pipeline_vlg_tst/Rd_in                xxxxxxxx
/pipeline_vlg_tst/WBData_EX            xxxxxxxx
/pipeline_vlg_tst/MemData_EX           xxxxxxxx
/pipeline_vlg_tst/MemWriteEn           0
/pipeline_vlg_tst/flash_EX             0
/pipeline_vlg_tst/flash_ID             0
/pipeline_vlg_tst/Immediate32_ID       xxxxxxxx
/pipeline_vlg_tst/Immediate32_IF       xxxxxxxx
/pipeline_vlg_tst/RegShift             xxxxxxxx
Now        3697000000000 ps
Cursor 1   418158806 ps
```

r14 并未有初值，其值是从 r9 转入内存再转入 r14，因此，add 的正常执行表示 sw 与 lw 都正常执行了。

lw,sw 及其相关指令的综合测试

state:

本项测试将 lwl,lwr,swl,swr 中每一种转存方式都测试了一次，详情请看以下测试

initial

register[0] = 32'h0;

register[1] = 32'h11112345;

register[2] = 32'h2;

register[3] = 32'h3;

register[4] = 32'h4;

register[5] = 32'h55556789;

register[8] = 32'h88;

register[9] = 32'h5467_8932;

register[10] = 32'h3476_8906;

register[11] = 32'hfffa_bcde;

register[12] = 32'h6789_3954;

register[13] = 32'h88;

register[30] = 32'hffff_ffff;

register[31] = 32'h7fff_ffff;

二进制代码：

ram[0] = 32'b0;

ram[1] = 32'b0;

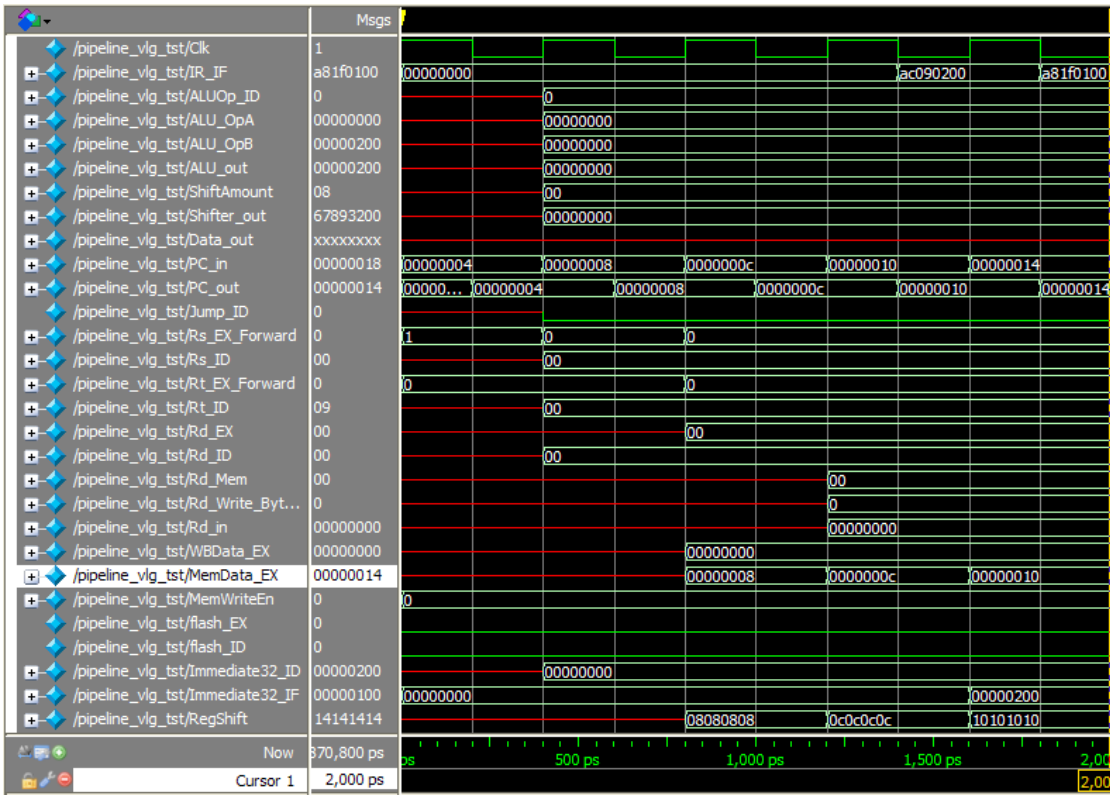
ram[2] = 32'b0;

ram[3] = 32'b0;

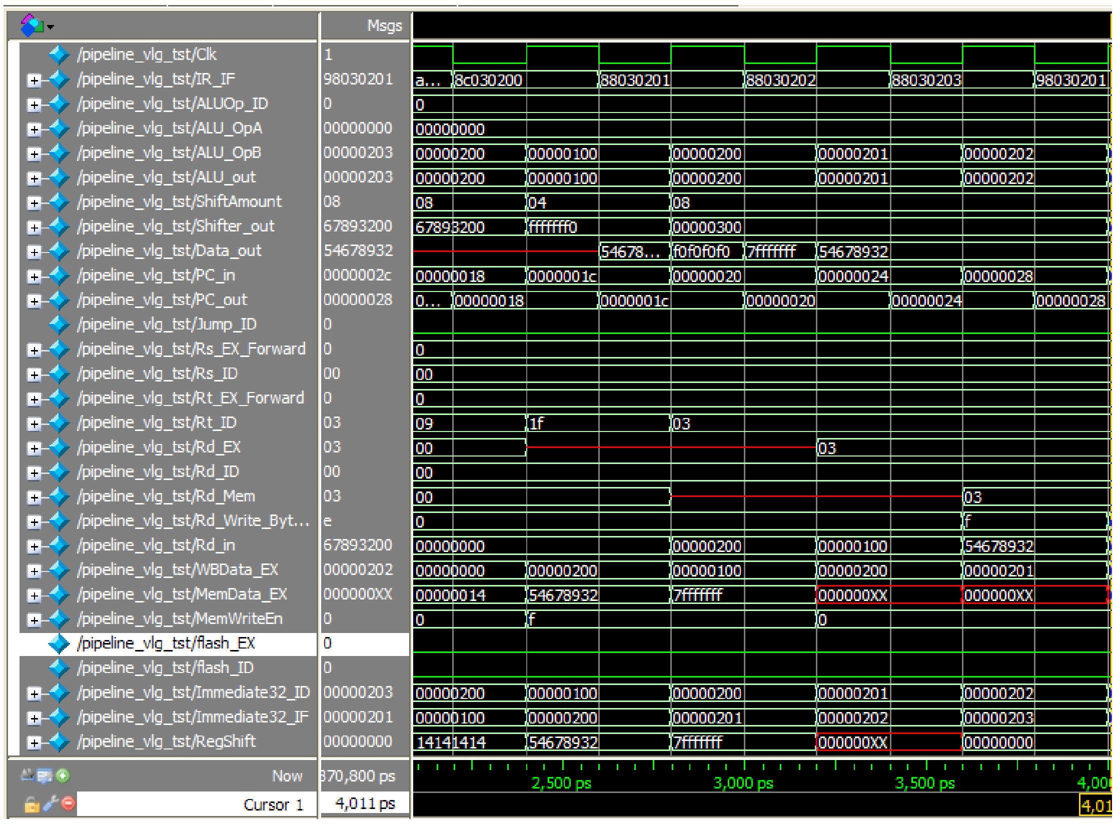
```
ram[4] = {16'b101011_00000_01001,16'd512};//sw    Reg[9] -> DM[512]
ram[5] = {16'b101010_00000_11111,16'd256};//swl   reg[31] -> DM[256]
ram[6] = {16'b100011_00000_00011,16'd512};//lw    ID[512]   ->   reg[3]
ram[7] = {16'b100010_00000_00011,16'd513};//lwl   DM[513] -> reg[3]
ram[8] = {16'b100010_00000_00011,16'd514};//lwl   DM[514] -> reg[3]
ram[9] = {16'b100010_00000_00011,16'd515};//lwl   DM[515] -> reg[3]
ram[10] = {16'b100110_00000_00011,16'd513};//lwr   DM[513] -> reg[3]
ram[11] = {16'b100110_00000_00011,16'd514};//lwr   DM[514] -> reg[3]
ram[12] = {16'b100110_00000_00011,16'd515};//lwr   DM[515] -> reg[3]
ram[13] = {16'b101010_00000_11111,16'd513};//swl   reg[31] -> DM[513]
ram[14] = {16'b101010_00000_11111,16'd514};//swl   reg[31] -> DM[514]
ram[15] = {16'b101010_00000_11111,16'd515};//swl   reg[31] -> DM[515]
ram[16] = {16'b101110_00000_11111,16'd513};//swr   reg[31] -> DM[513]
ram[17] = {16'b101110_00000_11111,16'd514};//swr   reg[31] -> DM[514]
ram[18] = {16'b101110_00000_11111,16'd515};//swr   reg[31] -> DM[515]
```

仿真截图：

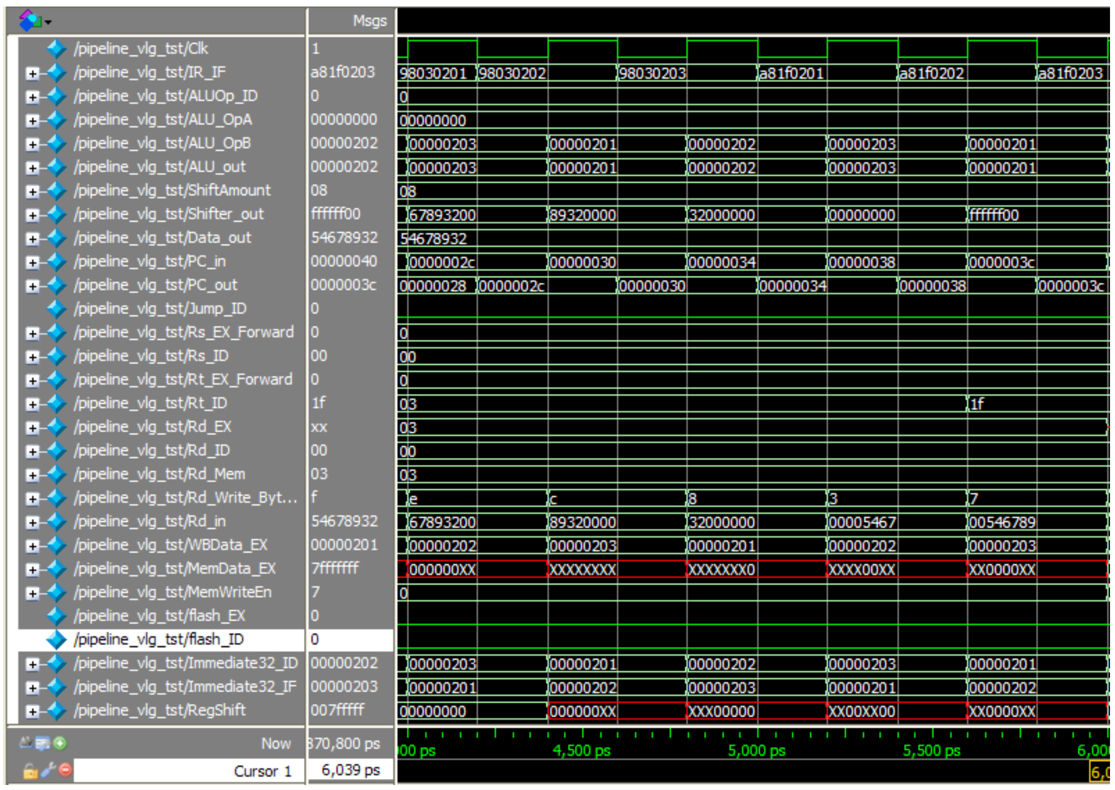
以下截图五个周期一张图，每张图无重复周期



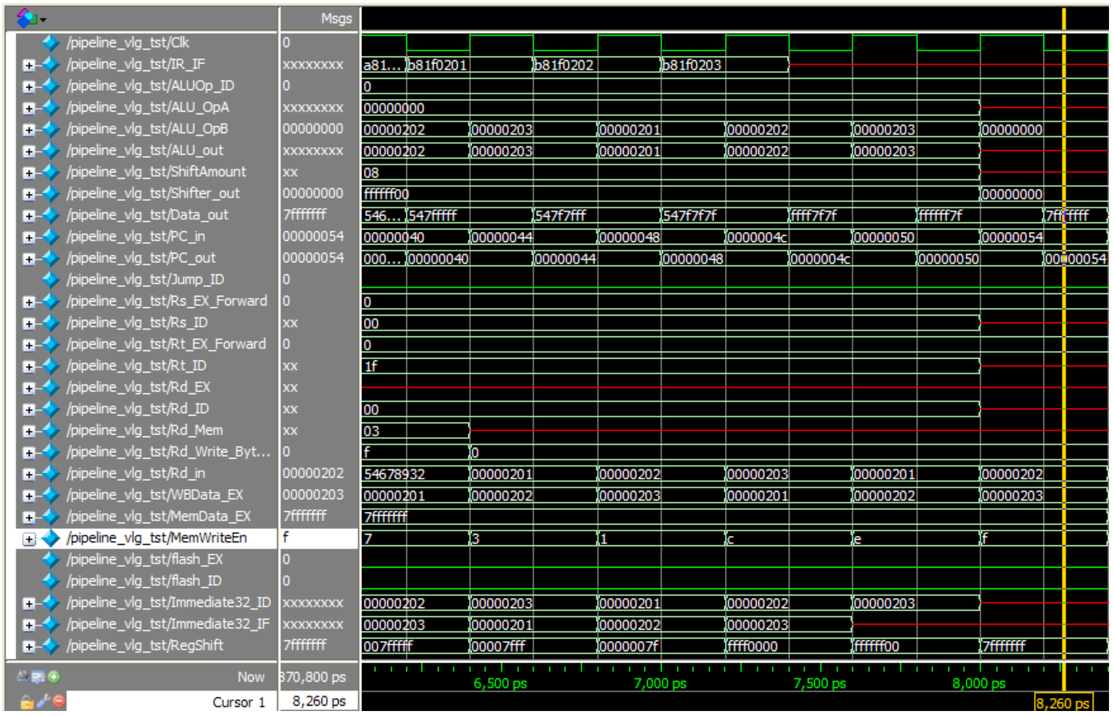
```
/pipeline_vlg_tst/Clk                1
/pipeline_vlg_tst/IR_IF              a81f0100
/pipeline_vlg_tst/ALUOp_ID           0
/pipeline_vlg_tst/ALU_OpA            00000000
/pipeline_vlg_tst/ALU_OpB            00000200
/pipeline_vlg_tst/ALU_out            00000200
/pipeline_vlg_tst/ShiftAmount        08
/pipeline_vlg_tst/Shifter_out        67893200
/pipeline_vlg_tst/Data_out           xxxxxxxx
/pipeline_vlg_tst/PC_in              00000018
/pipeline_vlg_tst/PC_out             00000014
/pipeline_vlg_tst/Jump_ID            0
/pipeline_vlg_tst/Rs_EX_Forward      0
/pipeline_vlg_tst/Rs_ID              00
/pipeline_vlg_tst/Rt_EX_Forward      0
/pipeline_vlg_tst/Rt_ID              09
/pipeline_vlg_tst/Rd_EX              00
/pipeline_vlg_tst/Rd_ID              00
/pipeline_vlg_tst/Rd_Mem             00
/pipeline_vlg_tst/Rd_Write_Byt...    0
/pipeline_vlg_tst/Rd_in              00000000
/pipeline_vlg_tst/WBData_EX          00000000
/pipeline_vlg_tst/MemData_EX         00000014
/pipeline_vlg_tst/MemWriteEn         0
/pipeline_vlg_tst/flash_EX           0
/pipeline_vlg_tst/flash_ID           0
/pipeline_vlg_tst/Immediate32_ID     00000200
/pipeline_vlg_tst/Immediate32_IF     00000100
/pipeline_vlg_tst/RegShift           14141414
Now      370,800 ps
Cursor 1   2,000 ps
```



	Msgs
/pipeline_vlg_tst/Clk	1
/pipeline_vlg_tst/IR_IF	98030201
/pipeline_vlg_tst/ALUOp_ID	0
/pipeline_vlg_tst/ALU_OpA	00000000
/pipeline_vlg_tst/ALU_OpB	00000203
/pipeline_vlg_tst/ALU_out	00000203
/pipeline_vlg_tst/ShiftAmount	08
/pipeline_vlg_tst/Shifter_out	67893200
/pipeline_vlg_tst/Data_out	54678932
/pipeline_vlg_tst/PC_in	0000002c
/pipeline_vlg_tst/PC_out	00000028
/pipeline_vlg_tst/Jump_ID	0
/pipeline_vlg_tst/Rs_EX_Forward	0
/pipeline_vlg_tst/Rs_ID	00
/pipeline_vlg_tst/Rt_EX_Forward	0
/pipeline_vlg_tst/Rt_ID	03
/pipeline_vlg_tst/Rd_EX	03
/pipeline_vlg_tst/Rd_ID	00
/pipeline_vlg_tst/Rd_Mem	03
/pipeline_vlg_tst/Rd_Write_Byt…	e
/pipeline_vlg_tst/Rd_in	67893200
/pipeline_vlg_tst/WBData_EX	00000202
/pipeline_vlg_tst/MemData_EX	000000XX
/pipeline_vlg_tst/MemWriteEn	0
/pipeline_vlg_tst/flash_EX	0
/pipeline_vlg_tst/flash_ID	0
/pipeline_vlg_tst/Immediate32_ID	00000203
/pipeline_vlg_tst/Immediate32_IF	00000201
/pipeline_vlg_tst/RegShift	00000000
Now	370,800 ps
Cursor 1	4,011 ps



	Msgs
/pipeline_vlg_tst/Clk	1
/pipeline_vlg_tst/IR_IF	a81f0203
/pipeline_vlg_tst/ALUOp_ID	0
/pipeline_vlg_tst/ALU_OpA	00000000
/pipeline_vlg_tst/ALU_OpB	00000202
/pipeline_vlg_tst/ALU_out	00000202
/pipeline_vlg_tst/ShiftAmount	08
/pipeline_vlg_tst/Shifter_out	ffffff00
/pipeline_vlg_tst/Data_out	54678932
/pipeline_vlg_tst/PC_in	00000040
/pipeline_vlg_tst/PC_out	0000003c
/pipeline_vlg_tst/Jump_ID	0
/pipeline_vlg_tst/Rs_EX_Forward	0
/pipeline_vlg_tst/Rs_ID	00
/pipeline_vlg_tst/Rt_EX_Forward	0
/pipeline_vlg_tst/Rt_ID	1f
/pipeline_vlg_tst/Rd_EX	xx
/pipeline_vlg_tst/Rd_ID	00
/pipeline_vlg_tst/Rd_Mem	03
/pipeline_vlg_tst/Rd_Write_Byt…	f
/pipeline_vlg_tst/Rd_in	54678932
/pipeline_vlg_tst/WBData_EX	00000201
/pipeline_vlg_tst/MemData_EX	7fffffff
/pipeline_vlg_tst/MemWriteEn	7
/pipeline_vlg_tst/flash_EX	0
/pipeline_vlg_tst/flash_ID	0
/pipeline_vlg_tst/Immediate32_ID	00000202
/pipeline_vlg_tst/Immediate32_IF	00000203
/pipeline_vlg_tst/RegShift	007fffff
Now	370,800 ps
Cursor 1	6,039 ps



	Msgs
/pipeline_vlg_tst/Clk	0
/pipeline_vlg_tst/IR_IF	xxxxxxxx
/pipeline_vlg_tst/ALUOp_ID	0
/pipeline_vlg_tst/ALU_OpA	xxxxxxxx
/pipeline_vlg_tst/ALU_OpB	00000000
/pipeline_vlg_tst/ALU_out	xxxxxxxx
/pipeline_vlg_tst/ShiftAmount	xx
/pipeline_vlg_tst/Shifter_out	00000000
/pipeline_vlg_tst/Data_out	7fffffff
/pipeline_vlg_tst/PC_in	00000054
/pipeline_vlg_tst/PC_out	00000054
/pipeline_vlg_tst/Jump_ID	0
/pipeline_vlg_tst/Rs_EX_Forward	0
/pipeline_vlg_tst/Rs_ID	xx
/pipeline_vlg_tst/Rt_EX_Forward	0
/pipeline_vlg_tst/Rt_ID	xx
/pipeline_vlg_tst/Rd_EX	xx
/pipeline_vlg_tst/Rd_ID	xx
/pipeline_vlg_tst/Rd_Mem	xx
/pipeline_vlg_tst/Rd_Write_Byt...	0
/pipeline_vlg_tst/Rd_in	00000202
/pipeline_vlg_tst/WBData_EX	00000203
/pipeline_vlg_tst/MemData_EX	7fffffff
/pipeline_vlg_tst/MemWriteEn	f
/pipeline_vlg_tst/flash_EX	0
/pipeline_vlg_tst/flash_ID	0
/pipeline_vlg_tst/Immediate32_ID	xxxxxxxx
/pipeline_vlg_tst/Immediate32_IF	xxxxxxxx
/pipeline_vlg_tst/RegShift	7fffffff
Now	370,800 ps
Cursor 1	8,260 ps

4、跳转及分支指令测试

initial:

register[0] = 0;

register[1] = 1;

register[2] = 5;

register[3] = 4;

register[4] = 3;

register[5] = 2;

register[8] = 32'hfffffffe;

register[9] = 9;

register[20] = 32'h7ffffffc;

register[21] = 32'h7ffffffd;

register[22] = 32'hfffffffe;

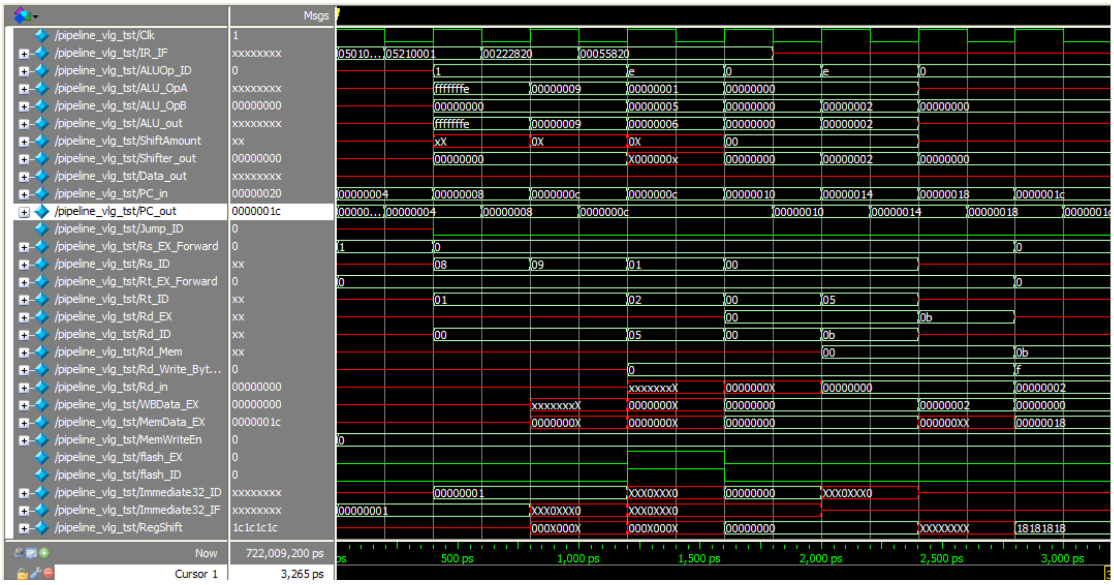
begz

ram[0] = 32'b000001_01000_00001_0000000000000001; //bgez r8 -> 1 跳转失败

ram[1] = 32'b000001_01001_00001_0000000000000001; //bgez r9 -> 1 跳转成功

ram[2] = 32'b000000_00001_00010_0010100000100000; //add r1 r2 = 6 -> r5

ram[3] = 32'b000000_00000_00101_0101100000100000; //add r0 r5 = ? -> r11


/pipeline_vlg_tst/Clk
/pipeline_vlg_tst/IR_IF
/pipeline_vlg_tst/ALUOp_ID
/pipeline_vlg_tst/ALU_OpA
/pipeline_vlg_tst/ALU_OpB
/pipeline_vlg_tst/ALU_out
/pipeline_vlg_tst/ShiftAmount
/pipeline_vlg_tst/Shifter_out
/pipeline_vlg_tst/Data_out
/pipeline_vlg_tst/PC_in
/pipeline_vlg_tst/PC_out
/pipeline_vlg_tst/Jump_ID
/pipeline_vlg_tst/Rs_EX_Forward
/pipeline_vlg_tst/Rs_ID
/pipeline_vlg_tst/Rt_EX_Forward
/pipeline_vlg_tst/Rt_ID
/pipeline_vlg_tst/Rd_EX
/pipeline_vlg_tst/Rd_ID
/pipeline_vlg_tst/Rd_Mem
/pipeline_vlg_tst/Rd_Write_Byt...
/pipeline_vlg_tst/Rd_in
/pipeline_vlg_tst/WBData_EX
/pipeline_vlg_tst/MemData_EX
/pipeline_vlg_tst/MemWriteEn
/pipeline_vlg_tst/flash_EX
/pipeline_vlg_tst/flash_ID
/pipeline_vlg_tst/Immediate32_ID
/pipeline_vlg_tst/Immediate32_IF
/pipeline_vlg_tst/RegShift
Now 722,009,200 ps
Cursor 1 3,265 ps
从图中可以看出，r0+r5 的值为 2，意味着跳过了 r5 加为 6 的指令操作，跳转有效

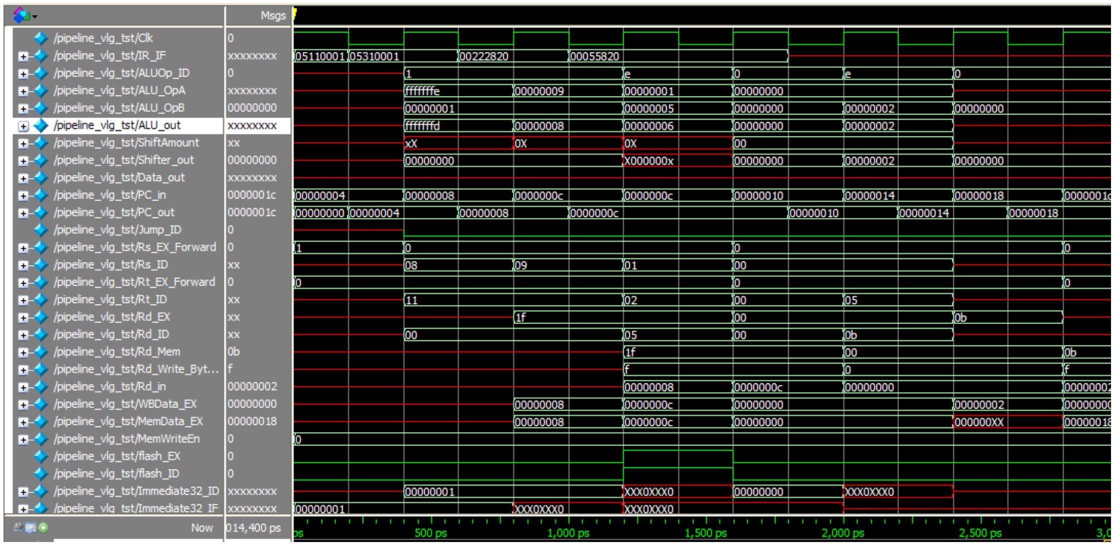
begzal

ram[0] = 32'b000001_01000_10001_0000000000000001;　//bgez r8 -> 1　跳转失败

ram[1] = 32'b000001_01001_10001_0000000000000001;　//bgez r9 -> 1　跳转成功

ram[2] = 32'b000000_00001_00010_0010100000100000;　//add　r1 r2 = 6 -> r5

ram[3] = 32'b000000_00000_00101_0101100000100000;　//add　r0 r5 = ? -> r11

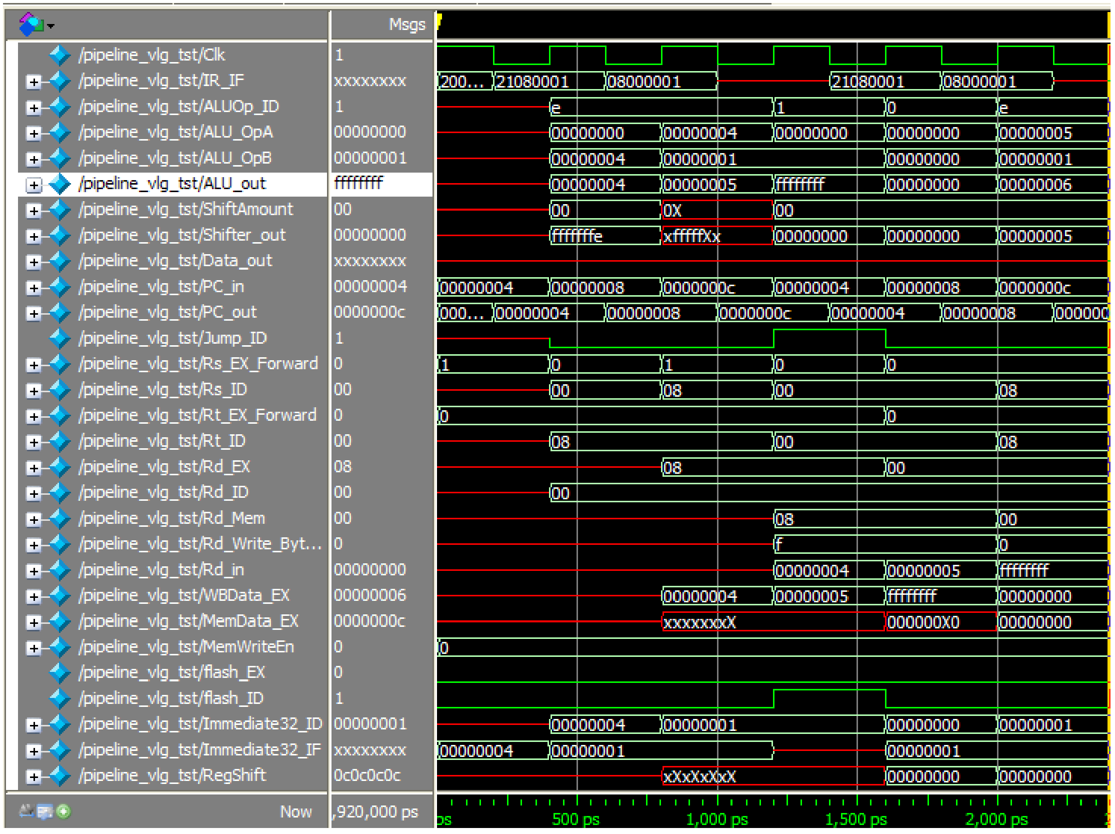

/pipeline_vlg_tst/Clk
/pipeline_vlg_tst/IR_IF
/pipeline_vlg_tst/ALUOp_ID
/pipeline_vlg_tst/ALU_OpA
/pipeline_vlg_tst/ALU_OpB
/pipeline_vlg_tst/ALU_out
/pipeline_vlg_tst/ShiftAmount
/pipeline_vlg_tst/Shifter_out
/pipeline_vlg_tst/Data_out
/pipeline_vlg_tst/PC_in
/pipeline_vlg_tst/PC_out
/pipeline_vlg_tst/Jump_ID
/pipeline_vlg_tst/Rs_EX_Forward
/pipeline_vlg_tst/Rs_ID
/pipeline_vlg_tst/Rt_EX_Forward
/pipeline_vlg_tst/Rt_ID
/pipeline_vlg_tst/Rd_EX
/pipeline_vlg_tst/Rd_ID
/pipeline_vlg_tst/Rd_Mem
/pipeline_vlg_tst/Rd_Write_Byt...
/pipeline_vlg_tst/Rd_in
/pipeline_vlg_tst/WBData_EX
/pipeline_vlg_tst/MemData_EX
/pipeline_vlg_tst/MemWriteEn
/pipeline_vlg_tst/flash_EX
/pipeline_vlg_tst/flash_ID
/pipeline_vlg_tst/Immediate32_ID
/pipeline_vlg_tst/Immediate32_IF
Now 014,400 ps
原理同 begz，r0+r5 的值为 2，begzal 有效

j

ram[0] = 32'b001000_00000_01000_0000000000000100;　//addi r0 im(4) -> r8

ram[1] = 32'b001000_01000_01000_0000000000000001;　//addi r8 im(1) -> r8

ram[2] = 32'b000010_00000_00000_0000000000000001; //j　　jump to 1


Msgs
/pipeline_vlg_tst/Clk 1
/pipeline_vlg_tst/IR_IF xxxxxxxx
/pipeline_vlg_tst/ALUOp_ID 1
/pipeline_vlg_tst/ALU_OpA 00000000
/pipeline_vlg_tst/ALU_OpB 00000001
/pipeline_vlg_tst/ALU_out ffffffff
/pipeline_vlg_tst/ShiftAmount 00
/pipeline_vlg_tst/Shifter_out 00000000
/pipeline_vlg_tst/Data_out xxxxxxxx
/pipeline_vlg_tst/PC_in 00000004
/pipeline_vlg_tst/PC_out 0000000c
/pipeline_vlg_tst/Jump_ID 1
/pipeline_vlg_tst/Rs_EX_Forward 0
/pipeline_vlg_tst/Rs_ID 00
/pipeline_vlg_tst/Rt_EX_Forward 0
/pipeline_vlg_tst/Rt_ID 00
/pipeline_vlg_tst/Rd_EX 08
/pipeline_vlg_tst/Rd_ID 00
/pipeline_vlg_tst/Rd_Mem 00
/pipeline_vlg_tst/Rd_Write_Byt... 0
/pipeline_vlg_tst/Rd_in 00000000
/pipeline_vlg_tst/WBData_EX 00000006
/pipeline_vlg_tst/MemData_EX 0000000c
/pipeline_vlg_tst/MemWriteEn 0
/pipeline_vlg_tst/flash_EX 0
/pipeline_vlg_tst/flash_ID 1
/pipeline_vlg_tst/Immediate32_ID 00000001
/pipeline_vlg_tst/Immediate32_IF xxxxxxxx
/pipeline_vlg_tst/RegShift 0c0c0c0c
Now ,920,000 ps
ps 500 ps 1,000 ps 1,500 ps 2,000 ps

从图中可以看出，r8 在不断的加一，因此 j 指令生效

beq:

ram[0] = 32'b001000_00000_00001_0000000000000001; //addi r0 im(1) -> r1

ram[1] = 32'b000000_00001_00001_0001000000100000; //add r1 r1 -> r2

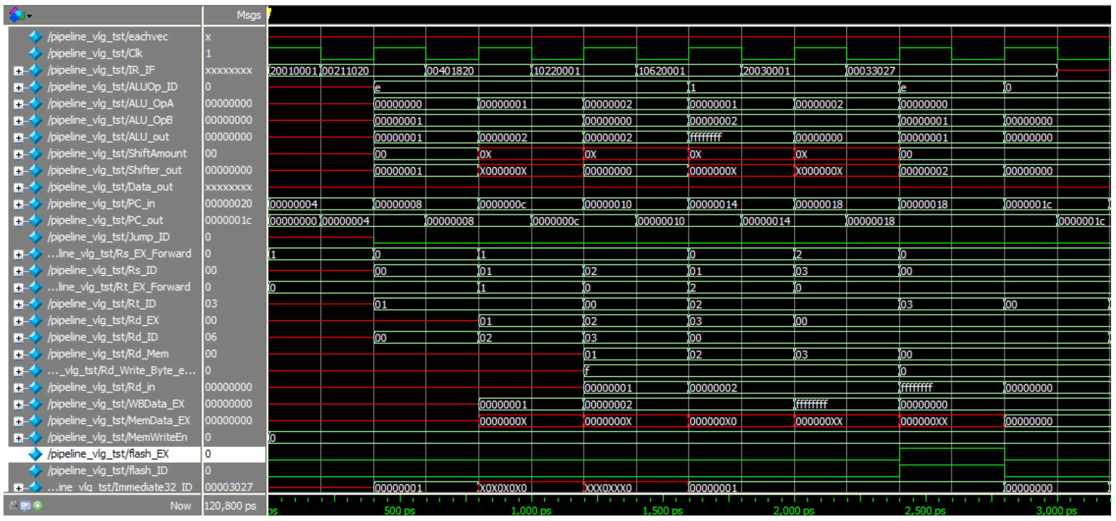
ram[2] = 32'b000000_00010_00000_0001100000100000; //add r2 r0 -> r3

ram[3] = 32'b000100_00001_00010_0000000000000001; //beq (r1 == r2)? ->1 跳转失败

ram[4] = 32'b000100_00011_00010_0000000000000001; //beq (r3 == r2)? ->1 跳转成功

ram[5] = 32'b001000_00000_00011_0000000000000001; //addi r0 im(1) -> r3

ram[6] = 32'b000000_00000_00011_00110_00000100111; //nor r0 r3 -> r6


Msgs
/pipeline_vlg_tst/eachvec x
/pipeline_vlg_tst/Clk 1
/pipeline_vlg_tst/IR_IF xxxxxxxx
/pipeline_vlg_tst/ALUOp_ID 0
/pipeline_vlg_tst/ALU_OpA 00000000
/pipeline_vlg_tst/ALU_OpB 00000000
/pipeline_vlg_tst/ALU_out 00000000
/pipeline_vlg_tst/ShiftAmount 00
/pipeline_vlg_tst/Shifter_out 00000000
/pipeline_vlg_tst/Data_out xxxxxxxx
/pipeline_vlg_tst/PC_in 00000020
/pipeline_vlg_tst/PC_out 0000001c
/pipeline_vlg_tst/Jump_ID 0
...line_vlg_tst/Rs_EX_Forward 0
/pipeline_vlg_tst/Rs_ID 00
...line_vlg_tst/Rt_EX_Forward 0
/pipeline_vlg_tst/Rt_ID 03
/pipeline_vlg_tst/Rd_EX 00
/pipeline_vlg_tst/Rd_ID 06
/pipeline_vlg_tst/Rd_Mem 00
..._vlg_tst/Rd_Write_Byte_e... 0
/pipeline_vlg_tst/Rd_in 00000000
/pipeline_vlg_tst/WBData_EX 00000000
/pipeline_vlg_tst/MemData_EX 00000000
/pipeline_vlg_tst/MemWriteEn 0
/pipeline_vlg_tst/flash_EX 0
/pipeline_vlg_tst/flash_ID 0
...ine_vlg_tst/Immediate32_ID 00003027
Now 120,800 ps
ps 500 ps 1,000 ps 1,500 ps 2,000 ps 2,500 ps 3,000 ps

conclusion：

经以上测试，单条指令的功能都已实现

二、冒险测试

test-1：

test-1 测试数据冒险，将 I-type R-type 指令的数据冒险均测试了一遍
每条指令中用的寄存器都是上一条指令更新的寄存器
test-1 的截图每张截图 5 个周期，不同截图的周期不重复

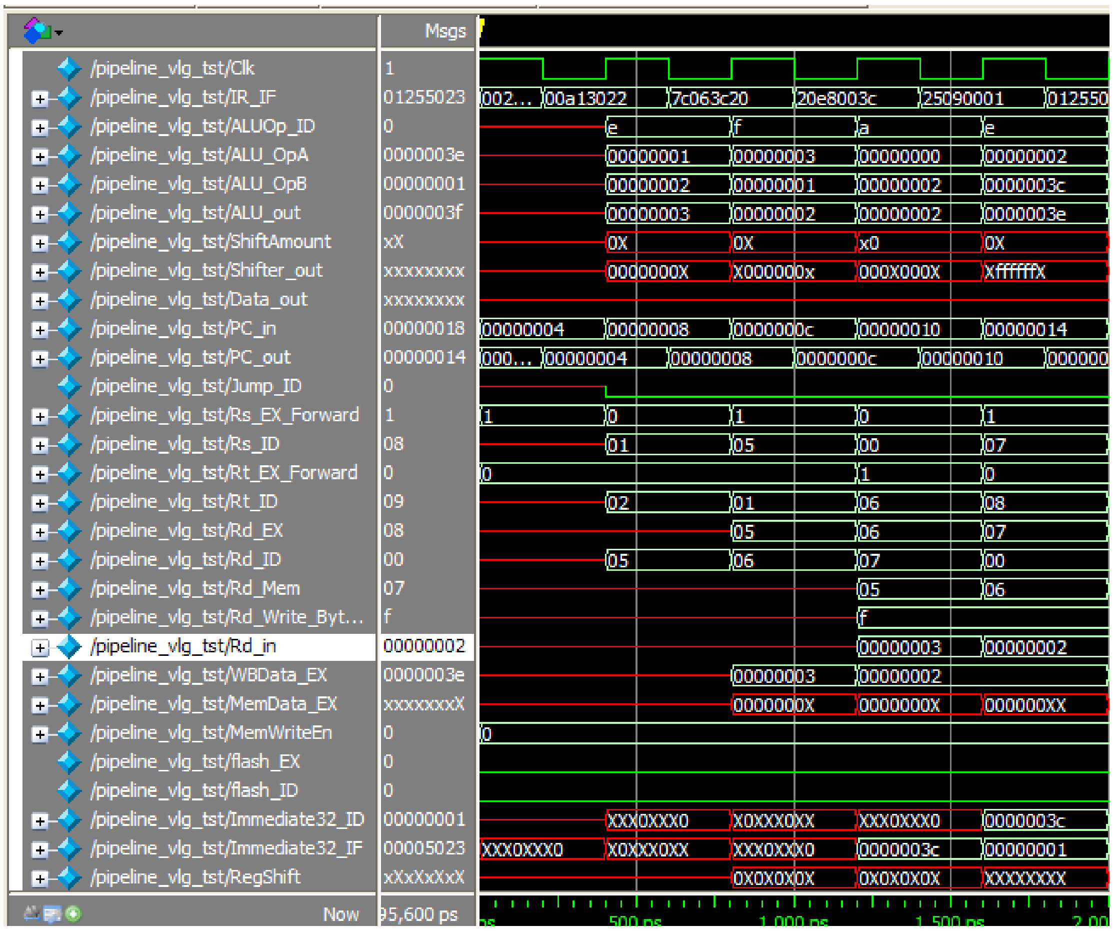
initial

```
    register[0] = 0;
    register[1] = 1;
    register[2] = 2;
```

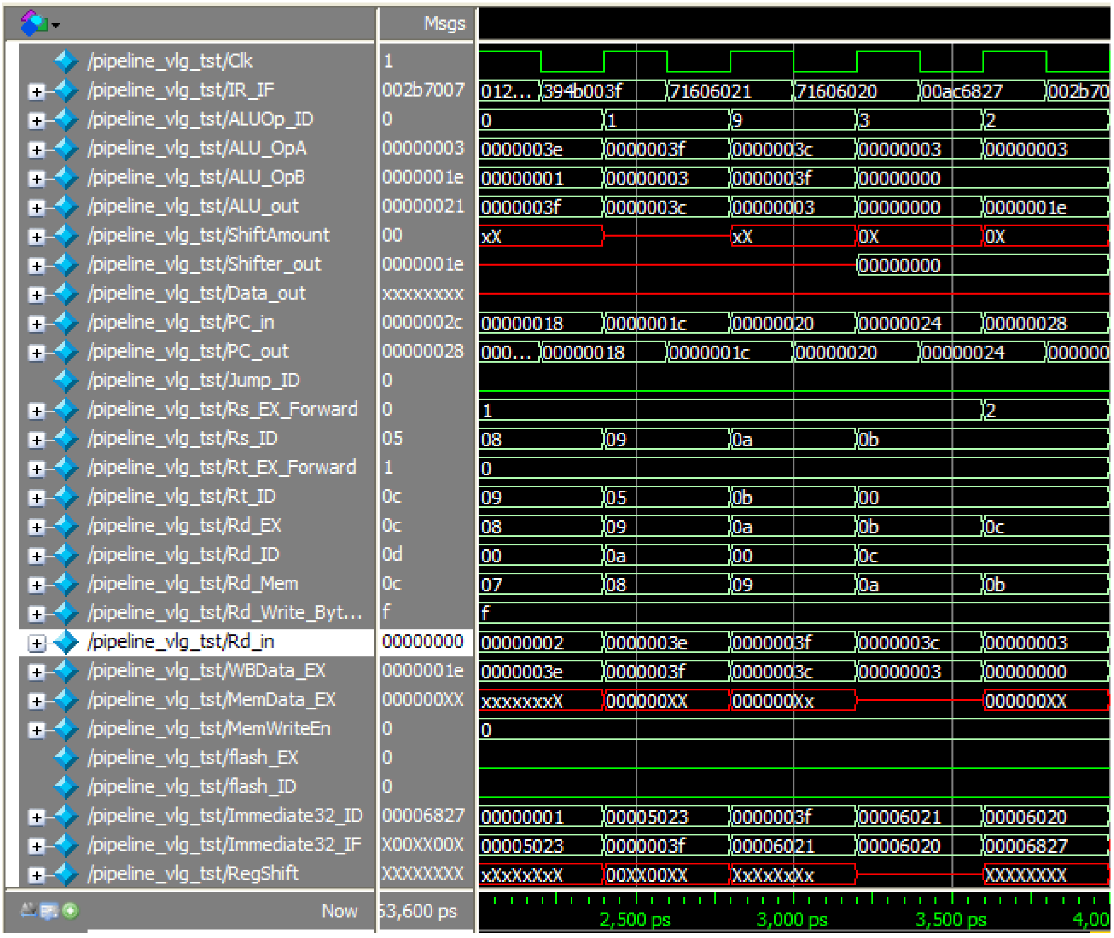
二进制代码：

```
    ram[0]=32'b000000_00001_00010_0010100000100000;    //add    r1 r2 = 3 -> r5
    ram[1]=32'b000000_00101_00001_0011000000100010;    //sub    r5 r1 = 2 -> r6
    ram[2]=32'b011111_00000_00110_0011110000100000;    //seb    r6        = 2 -> r7
    ram[3]=32'b001000_00111_01000_0000000000111100;    //addi r7 im = 111110 -> r8  溢
    ram[4]=32'b001001_01000_01001_0000000000000001;    //addiu r8 im = 111111 -> r9
    ram[5]=32'b000000_01001_00101_0101000000100011;    //subu r9 r5 = 111100 -> r10
    ram[6]=32'b001110_01010_01011_0000000000111111;    //xori r10 im = 3 -> r11
    ram[7]=32'b011100_01011_00000_0110000000100001;    //clo r11 = 0 -> r12
    ram[8]=32'b011100_01011_00000_0110000000100000;    //clz r11 = 30 -> r12
    ram[9]=32'b000000_00101_01100_0110100000100111;    //nor r5 r12 -> r13
    ram[10]=32'b000000_00001_01011_0111000000000111; //srav r1 r11 -> r14
    ram[11]=32'b000000_00001_01110_0111100100000010; //rotr r14 4 -> r15
    ram[12]=32'b000000_01111_01110_1000000000101011; //sltu r15 r14 -> r16
    ram[13]=32'b001010_10000_10000_0111111111111111; //slti r16 0x7fff -> r16
```

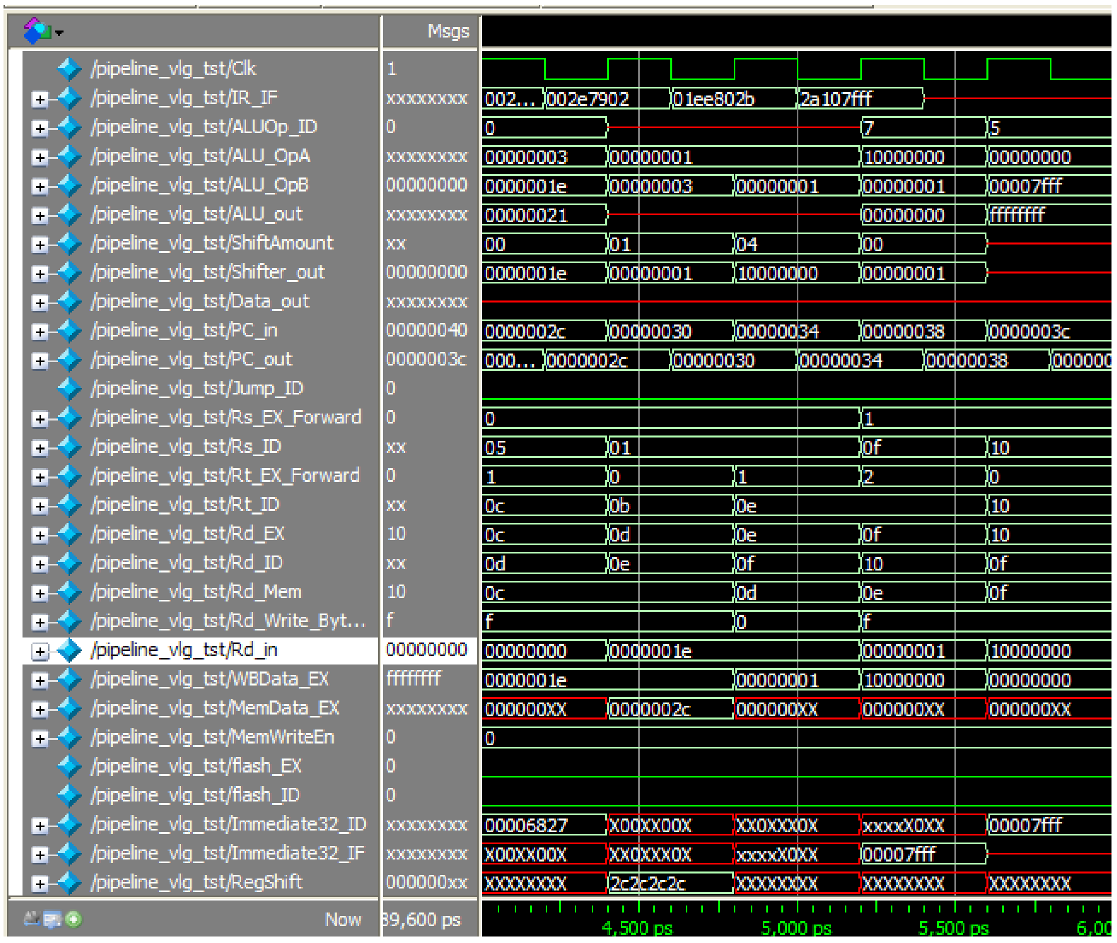
仿真截图：



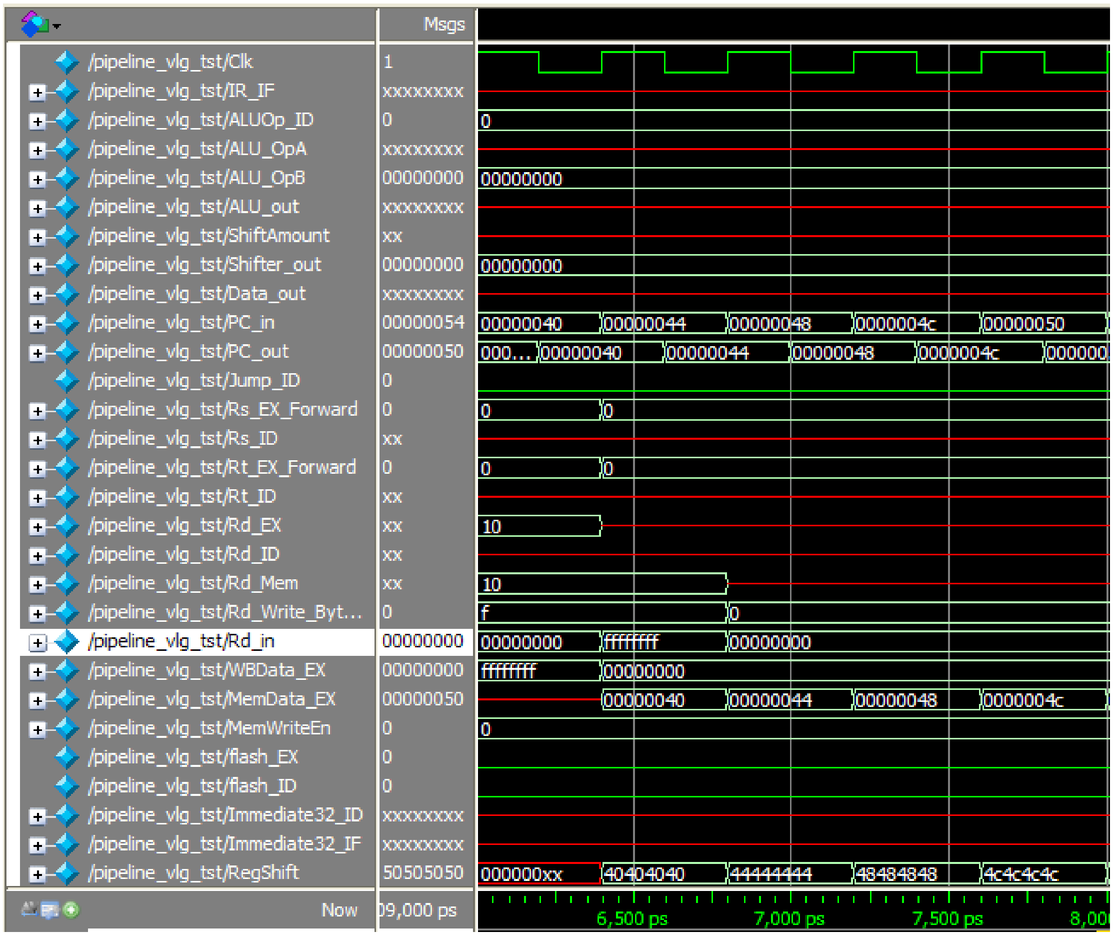
```
                                    Msgs
/pipeline_vlg_tst/Clk               1
/pipeline_vlg_tst/IR_IF             01255023   002... 00a13022 7c063c20 20e8003c 25090001 012550
/pipeline_vlg_tst/ALUOp_ID          0          e f a e
/pipeline_vlg_tst/ALU_OpA           0000003e   00000001 00000003 00000000 00000002
/pipeline_vlg_tst/ALU_OpB           00000001   00000002 00000001 00000002 0000003c
/pipeline_vlg_tst/ALU_out           0000003f   00000003 00000002 00000002 0000003e
/pipeline_vlg_tst/ShiftAmount       xX         0X 0X x0 0X
/pipeline_vlg_tst/Shifter_out       xxxxxxxx   0000000X X000000x 000X000X xffffffX
/pipeline_vlg_tst/Data_out          xxxxxxxx
/pipeline_vlg_tst/PC_in             00000018   00000004 00000008 0000000c 00000010 00000014
/pipeline_vlg_tst/PC_out            00000014   000... 00000004 00000008 0000000c 00000010 000000
/pipeline_vlg_tst/Jump_ID           0
/pipeline_vlg_tst/Rs_EX_Forward     1          1 0 1 0 1
/pipeline_vlg_tst/Rs_ID             08         01 05 00 07
/pipeline_vlg_tst/Rt_EX_Forward     0          0 1 0
/pipeline_vlg_tst/Rt_ID             09         02 01 06 08
/pipeline_vlg_tst/Rd_EX             08         05 06 07
/pipeline_vlg_tst/Rd_ID             00         05 06 07 00
/pipeline_vlg_tst/Rd_Mem            07         05 06
/pipeline_vlg_tst/Rd_Write_Byt...   f          f
/pipeline_vlg_tst/Rd_in             00000002   00000003 00000002
/pipeline_vlg_tst/WBData_EX         0000003e   00000003 00000002
/pipeline_vlg_tst/MemData_EX        xxxxxxxX   0000000X 0000000X 000000XX
/pipeline_vlg_tst/MemWriteEn        0          0
/pipeline_vlg_tst/flash_EX          0
/pipeline_vlg_tst/flash_ID          0
/pipeline_vlg_tst/Immediate32_ID    00000001   XXX0XXX0 X0XXX0XX XXX0XXX0 0000003c
/pipeline_vlg_tst/Immediate32_IF    00005023   XXX0XXX0 X0XXX0XX XXX0XXX0 0000003c 00000001
/pipeline_vlg_tst/RegShift          xXxXxXxX   0X0X0X0X 0X0X0X0X XXXXXXXX
Now                                 95,600 ps  ns  500 ps  1,000 ps  1,500 ps  2,00
```



```
                                    Msgs
/pipeline_vlg_tst/Clk               1
/pipeline_vlg_tst/IR_IF             002b7007   012... 394b003f 71606021 71606020 00ac6827 002b70
/pipeline_vlg_tst/ALUOp_ID          0          0 1 9 3 2
/pipeline_vlg_tst/ALU_OpA           00000003   0000003e 0000003f 0000003c 00000003 00000003
/pipeline_vlg_tst/ALU_OpB           0000001e   00000001 00000003 0000003f 00000000
/pipeline_vlg_tst/ALU_out           00000021   0000003f 0000003c 00000003 00000000 0000001e
/pipeline_vlg_tst/ShiftAmount       00         xX xX 0X 0X
/pipeline_vlg_tst/Shifter_out       0000001e   00000000
/pipeline_vlg_tst/Data_out          xxxxxxxx
/pipeline_vlg_tst/PC_in             0000002c   00000018 0000001c 00000020 00000024 00000028
/pipeline_vlg_tst/PC_out            00000028   000... 00000018 0000001c 00000020 00000024 000000
/pipeline_vlg_tst/Jump_ID           0
/pipeline_vlg_tst/Rs_EX_Forward     0          1 2
/pipeline_vlg_tst/Rs_ID             05         08 09 0a 0b
/pipeline_vlg_tst/Rt_EX_Forward     1          0
/pipeline_vlg_tst/Rt_ID             0c         09 05 0b 00
/pipeline_vlg_tst/Rd_EX             0c         08 09 0a 0b 0c
/pipeline_vlg_tst/Rd_ID             0d         00 0a 00 0c
/pipeline_vlg_tst/Rd_Mem            0c         07 08 09 0a 0b
/pipeline_vlg_tst/Rd_Write_Byt...   f          f
/pipeline_vlg_tst/Rd_in             00000000   00000002 0000003e 0000003f 0000003c 00000003
/pipeline_vlg_tst/WBData_EX         0000001e   0000003e 0000003f 0000003c 00000003 00000000
/pipeline_vlg_tst/MemData_EX        000000XX   xxxxxxxX 000000XX 000000Xx 000000XX
/pipeline_vlg_tst/MemWriteEn        0          0
/pipeline_vlg_tst/flash_EX          0
/pipeline_vlg_tst/flash_ID          0
/pipeline_vlg_tst/Immediate32_ID    00006827   00000001 00005023 0000003f 00006021 00006020
/pipeline_vlg_tst/Immediate32_IF    X00XX00X   00005023 0000003f 00006021 00006020 00006827
/pipeline_vlg_tst/RegShift          XXXXXXXX   xXxXxXxX 00XX00XX XxXxXxXx XXXXXXXX
Now                                 53,600 ps  2,500 ps  3,000 ps  3,500 ps  4,00
```



```
Msgs
/pipeline_vlg_tst/Clk                1
/pipeline_vlg_tst/IR_IF              xxxxxxxx   002... 002e7902 01ee802b 2a107fff
/pipeline_vlg_tst/ALUOp_ID           0          0 7 5
/pipeline_vlg_tst/ALU_OpA            xxxxxxxx   00000003 00000001 10000000 00000000
/pipeline_vlg_tst/ALU_OpB            00000000   0000001e 00000003 00000001 00000001 00007fff
/pipeline_vlg_tst/ALU_out            xxxxxxxx   00000021 00000000 ffffffff
/pipeline_vlg_tst/ShiftAmount        xx         00 01 04 00
/pipeline_vlg_tst/Shifter_out        00000000   0000001e 00000001 10000000 00000001
/pipeline_vlg_tst/Data_out           xxxxxxxx
/pipeline_vlg_tst/PC_in              00000040   0000002c 00000030 00000034 00000038 0000003c
/pipeline_vlg_tst/PC_out             0000003c   000... 0000002c 00000030 00000034 00000038 000000
/pipeline_vlg_tst/Jump_ID            0
/pipeline_vlg_tst/Rs_EX_Forward      0          0 1
/pipeline_vlg_tst/Rs_ID              xx         05 01 0f 10
/pipeline_vlg_tst/Rt_EX_Forward      0          1 0 1 2 0
/pipeline_vlg_tst/Rt_ID              xx         0c 0b 0e 10
/pipeline_vlg_tst/Rd_EX              10         0c 0d 0e 0f 10
/pipeline_vlg_tst/Rd_ID              xx         0d 0e 0f 10 0f
/pipeline_vlg_tst/Rd_Mem             10         0c 0d 0e 0f
/pipeline_vlg_tst/Rd_Write_Byt...    f          f 0 f
/pipeline_vlg_tst/Rd_in              00000000   00000000 0000001e 00000001 10000000
/pipeline_vlg_tst/WBData_EX          ffffffff   0000001e 00000001 10000000 00000000
/pipeline_vlg_tst/MemData_EX         xxxxxxxx   000000XX 0000002c 000000XX 000000XX 000000XX
/pipeline_vlg_tst/MemWriteEn         0          0
/pipeline_vlg_tst/flash_EX           0
/pipeline_vlg_tst/flash_ID           0
/pipeline_vlg_tst/Immediate32_ID     xxxxxxxx   00006827 X00XX00X XX0XXX0X xxxxX0XX 00007fff
/pipeline_vlg_tst/Immediate32_IF     xxxxxxxx   X00XX00X XX0XXX0X xxxxX0XX 00007fff
/pipeline_vlg_tst/RegShift           000000xx   XXXXXXXX 2c2c2c2c XXXXXXXX XXXXXXXX XXXXXXXX
Now   39,600 ps        4,500 ps   5,000 ps   5,500 ps   6,00
```



```
Msgs
/pipeline_vlg_tst/Clk                1
/pipeline_vlg_tst/IR_IF              xxxxxxxx
/pipeline_vlg_tst/ALUOp_ID           0          0
/pipeline_vlg_tst/ALU_OpA            xxxxxxxx
/pipeline_vlg_tst/ALU_OpB            00000000   00000000
/pipeline_vlg_tst/ALU_out            xxxxxxxx
/pipeline_vlg_tst/ShiftAmount        xx
/pipeline_vlg_tst/Shifter_out        00000000   00000000
/pipeline_vlg_tst/Data_out           xxxxxxxx
/pipeline_vlg_tst/PC_in              00000054   00000040 00000044 00000048 0000004c 00000050
/pipeline_vlg_tst/PC_out             00000050   000... 00000040 00000044 00000048 0000004c 000000
/pipeline_vlg_tst/Jump_ID            0
/pipeline_vlg_tst/Rs_EX_Forward      0          0 0
/pipeline_vlg_tst/Rs_ID              xx
/pipeline_vlg_tst/Rt_EX_Forward      0          0 0
/pipeline_vlg_tst/Rt_ID              xx
/pipeline_vlg_tst/Rd_EX              xx         10
/pipeline_vlg_tst/Rd_ID              xx
/pipeline_vlg_tst/Rd_Mem             xx         10
/pipeline_vlg_tst/Rd_Write_Byt...    0          f 0
/pipeline_vlg_tst/Rd_in              00000000   00000000 ffffffff 00000000
/pipeline_vlg_tst/WBData_EX          00000000   ffffffff 00000000
/pipeline_vlg_tst/MemData_EX         00000050   00000040 00000044 00000048 0000004c
/pipeline_vlg_tst/MemWriteEn         0          0
/pipeline_vlg_tst/flash_EX           0
/pipeline_vlg_tst/flash_ID           0
/pipeline_vlg_tst/Immediate32_ID     xxxxxxxx
/pipeline_vlg_tst/Immediate32_IF     xxxxxxxx
/pipeline_vlg_tst/RegShift           50505050   000000xx 40404040 44444444 48484848 4c4c4c4c
Now   39,000 ps        6,500 ps   7,000 ps   7,500 ps   8,00
```

test-1-conclusion：

经以上测试，I-type R-type 指令的数据冒险都无误

test-2：

本次测试进行利用多次跳转及分支完成了一个 for 循环，可完成连续跳转的冒险测试，以及数据冒险和控制冒险夹杂并且同时发成的测试。

汇编代码的意义是：将 r1 加为 1，r2 加为 2，r3 加为 3，r5 加为 0，然后进入循环。

循环做的是：begzal 跳过 begz，然后下面每次使 r5 加一，直到 r5 大于等于 r2，就使得 begzal 不执行，然后执行 begz，跳过 jump，结束循环。最后五个 add 是希望得到 r1~r5 的值，来看是否和预期值相同。

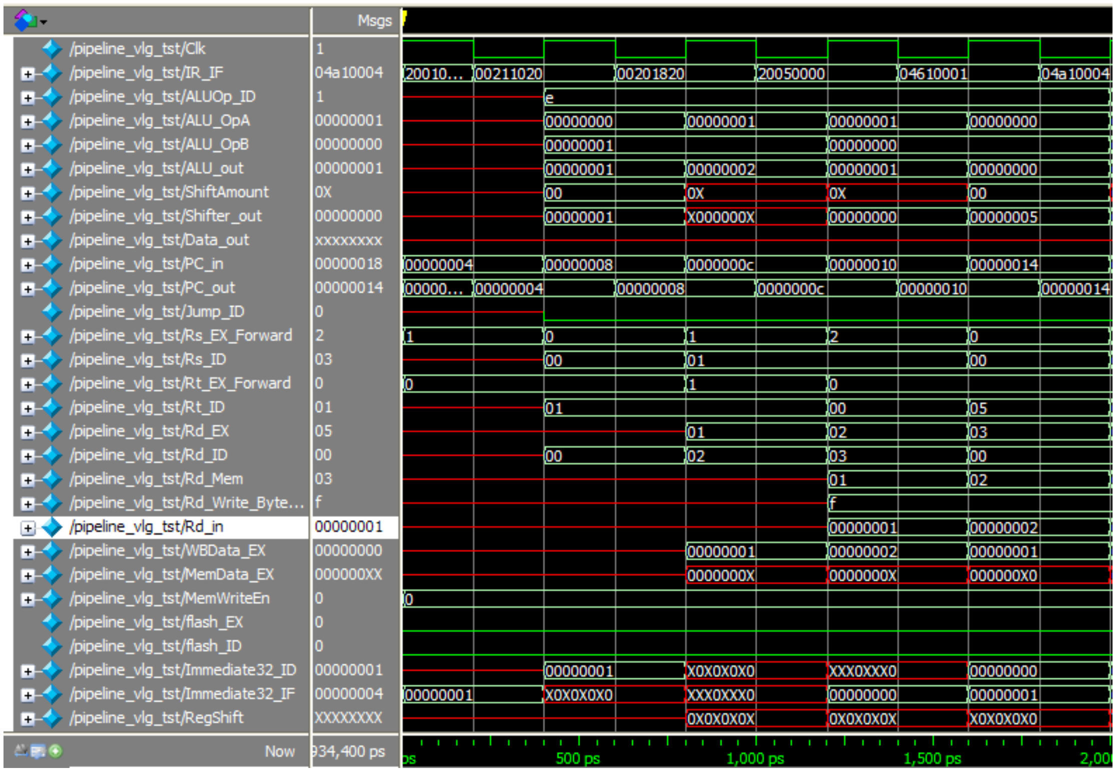
initial：

本次测试寄存器全无初始化，仅用到 r0 = 0；

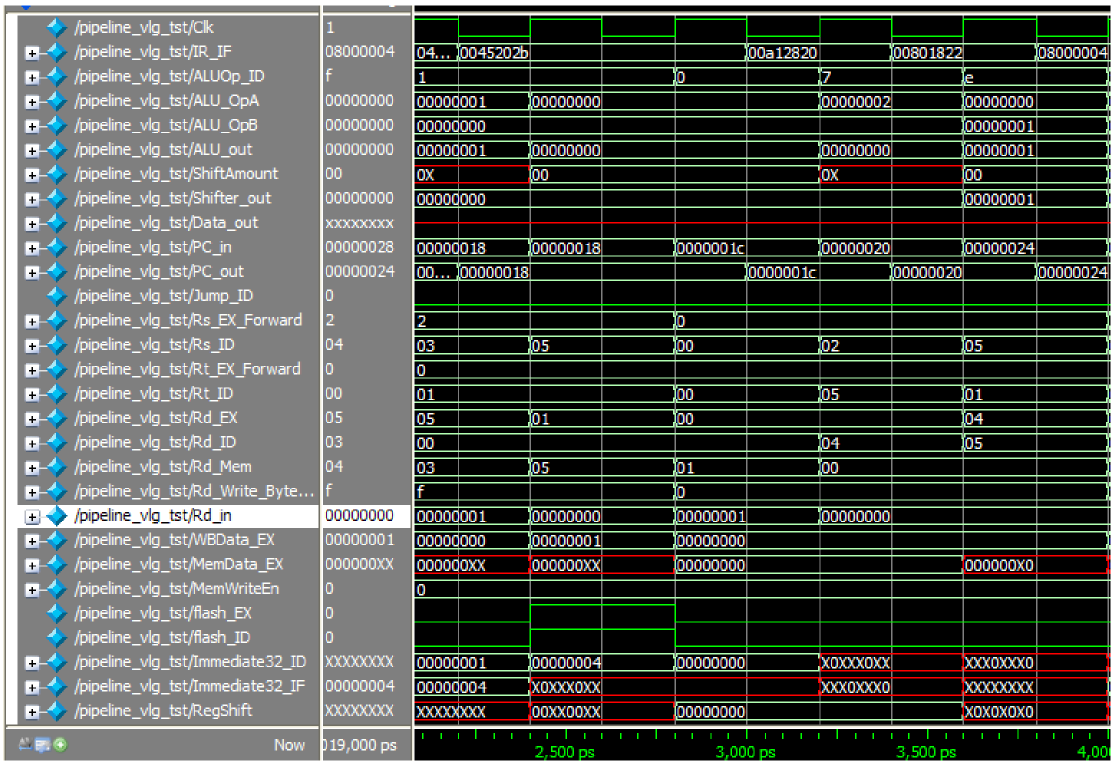
二进制代码：

```
ram[0] = 32'b00100000000000010000000000000001;    //addi    r0 im(1) -> r1
ram[1] = 32'b00000000001000010001000000100000;    //add      r1 r1 -> r2
ram[2] = 32'b00000000001000000001100000100000;    //add      r1 r0 -> r3
ram[3] = 32'b00100000000001010000000000000000;    //addi    r0    im(0) -> r5
ram[4] = 32'b00000100011100010000000000000001;    //begzal    r3 -> 1
ram[5] = 32'b00000100101000010000000000000100;    //begz      r5 -> 4
ram[6] = 32'b00000000101000100010000000101011;    //sltu    ( r5 r2 ) r4
ram[7] = 32'b00000000101000010010100000100000;    //add    r5    r1 -> r5
ram[8] = 32'b00000000100000010010100000100010;    //sub    r4    r1 ->r3
ram[9] = 32'b00001000000000000000000000000100;    //j    to    6*4
//    r1=1    r2=2    r3 =-1    r4=-1    r5 =4
ram[10] = 32'b000000_00000_00001_00110_00000100000; //add    r0 r1 -> r6
ram[11] = 32'b000000_00000_00010_00110_00000100000; //add    r0 r2 -> r6
ram[12] = 32'b000000_00000_00011_00110_00000100000; //add    r0 r3 -> r6
ram[13] = 32'b000000_00000_00100_00110_00000100000; //add    r0 r4 -> r6
ram[14] = 32'b000000_00000_00101_00110_00000100000; //add    r0 r5 -> r6
```

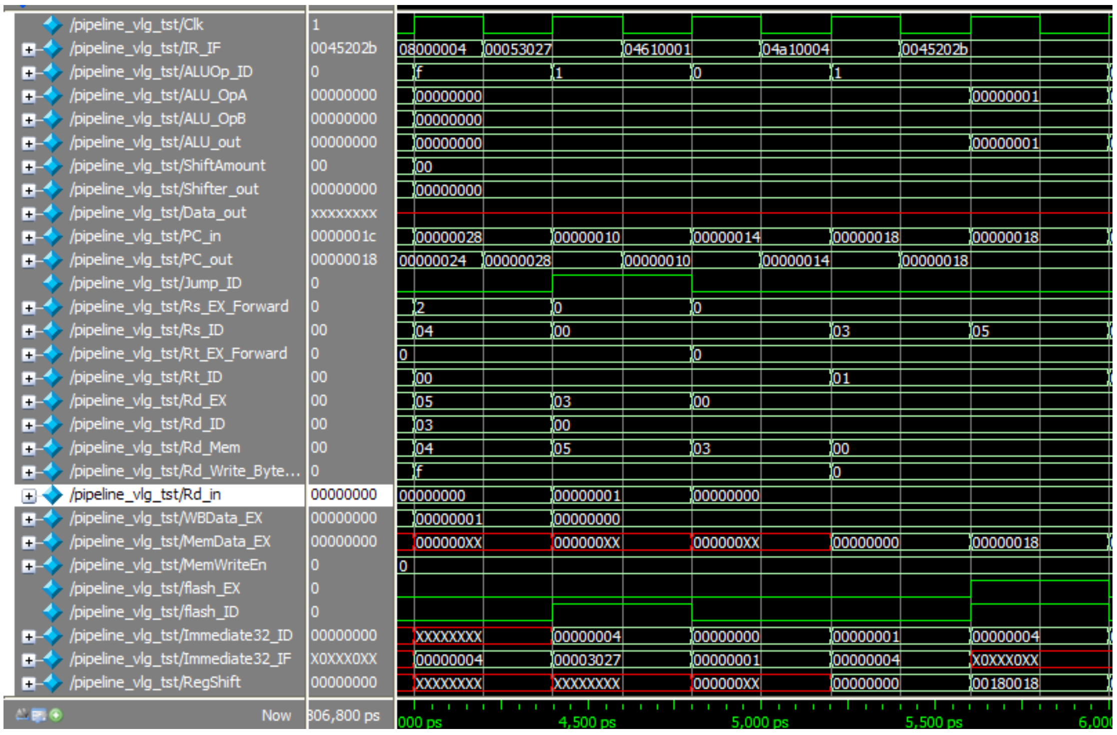
仿真截图：



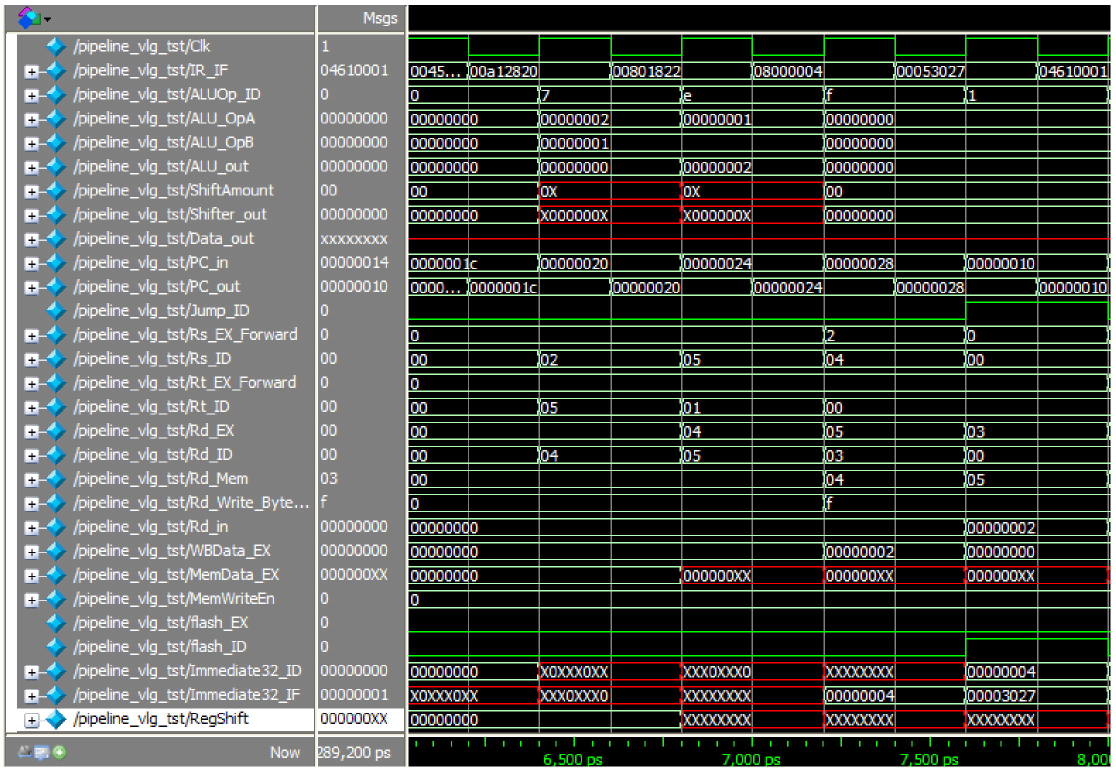
	Msgs
/pipeline_vlg_tst/Clk	1
/pipeline_vlg_tst/IR_IF	04a10004
/pipeline_vlg_tst/ALUOp_ID	1
/pipeline_vlg_tst/ALU_OpA	00000001
/pipeline_vlg_tst/ALU_OpB	00000000
/pipeline_vlg_tst/ALU_out	00000001
/pipeline_vlg_tst/ShiftAmount	0X
/pipeline_vlg_tst/Shifter_out	00000000
/pipeline_vlg_tst/Data_out	xxxxxxxx
/pipeline_vlg_tst/PC_in	00000018
/pipeline_vlg_tst/PC_out	00000014
/pipeline_vlg_tst/Jump_ID	0
/pipeline_vlg_tst/Rs_EX_Forward	2
/pipeline_vlg_tst/Rs_ID	03
/pipeline_vlg_tst/Rt_EX_Forward	0
/pipeline_vlg_tst/Rt_ID	01
/pipeline_vlg_tst/Rd_EX	05
/pipeline_vlg_tst/Rd_ID	00
/pipeline_vlg_tst/Rd_Mem	03
/pipeline_vlg_tst/Rd_Write_Byte...	f
/pipeline_vlg_tst/Rd_in	00000001
/pipeline_vlg_tst/WBData_EX	00000000
/pipeline_vlg_tst/MemData_EX	000000XX
/pipeline_vlg_tst/MemWriteEn	0
/pipeline_vlg_tst/flash_EX	0
/pipeline_vlg_tst/flash_ID	0
/pipeline_vlg_tst/Immediate32_ID	00000001
/pipeline_vlg_tst/Immediate32_IF	00000004
/pipeline_vlg_tst/RegShift	XXXXXXXX
Now	934,400 ps



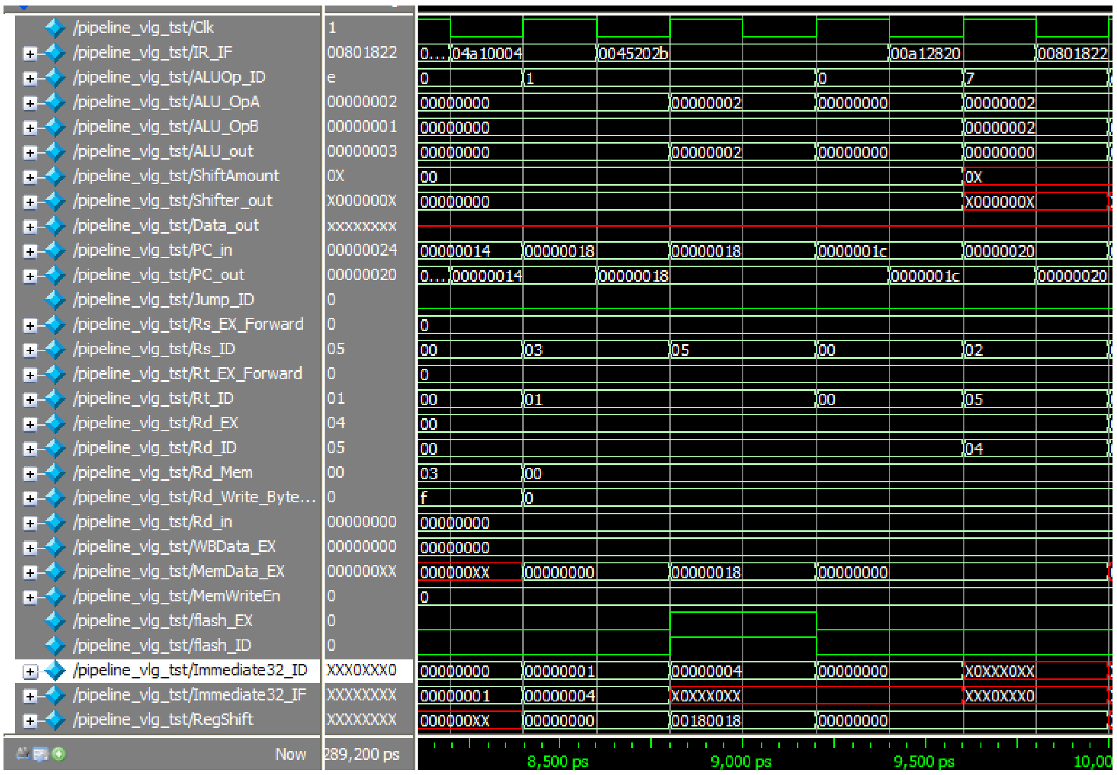
/pipeline_vlg_tst/Clk	1
/pipeline_vlg_tst/IR_IF	08000004
/pipeline_vlg_tst/ALUOp_ID	f
/pipeline_vlg_tst/ALU_OpA	00000000
/pipeline_vlg_tst/ALU_OpB	00000000
/pipeline_vlg_tst/ALU_out	00000000
/pipeline_vlg_tst/ShiftAmount	00
/pipeline_vlg_tst/Shifter_out	00000000
/pipeline_vlg_tst/Data_out	xxxxxxxx
/pipeline_vlg_tst/PC_in	00000028
/pipeline_vlg_tst/PC_out	00000024
/pipeline_vlg_tst/Jump_ID	0
/pipeline_vlg_tst/Rs_EX_Forward	2
/pipeline_vlg_tst/Rs_ID	04
/pipeline_vlg_tst/Rt_EX_Forward	0
/pipeline_vlg_tst/Rt_ID	00
/pipeline_vlg_tst/Rd_EX	05
/pipeline_vlg_tst/Rd_ID	03
/pipeline_vlg_tst/Rd_Mem	04
/pipeline_vlg_tst/Rd_Write_Byte...	f
/pipeline_vlg_tst/Rd_in	00000000
/pipeline_vlg_tst/WBData_EX	00000001
/pipeline_vlg_tst/MemData_EX	000000XX
/pipeline_vlg_tst/MemWriteEn	0
/pipeline_vlg_tst/flash_EX	0
/pipeline_vlg_tst/flash_ID	0
/pipeline_vlg_tst/Immediate32_ID	XXXXXXXX
/pipeline_vlg_tst/Immediate32_IF	00000004
/pipeline_vlg_tst/RegShift	XXXXXXXX
Now	019,000 ps



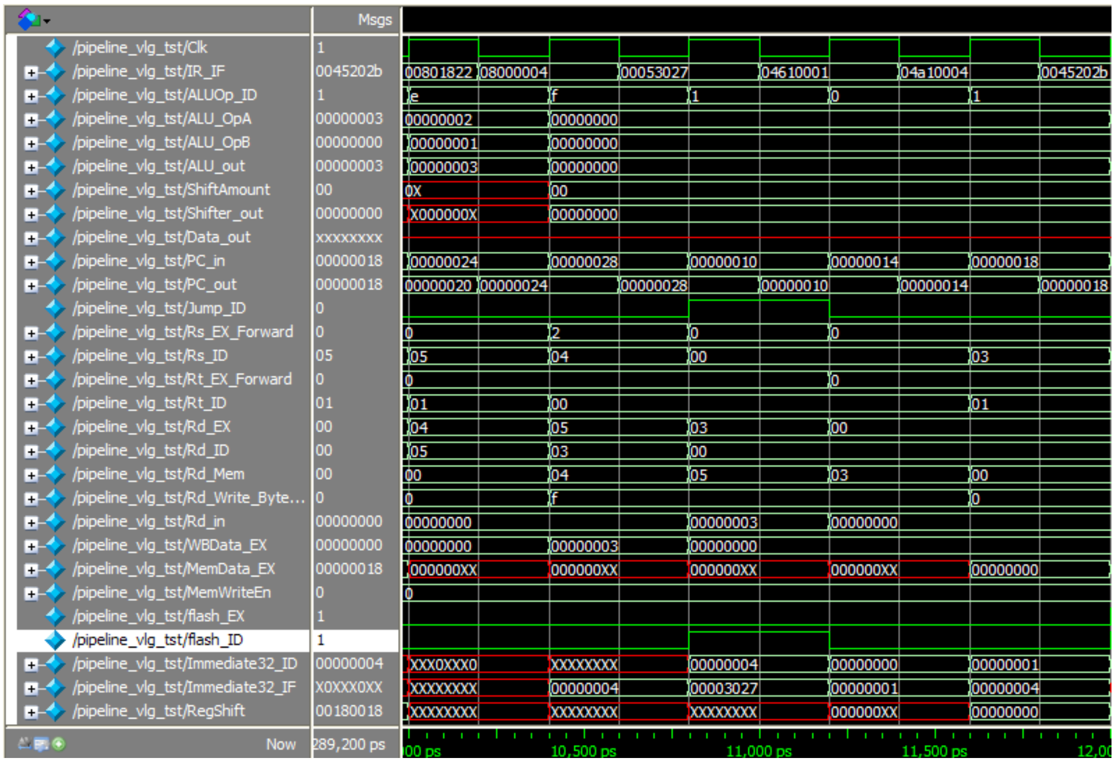
/pipeline_vlg_tst/Clk	1
/pipeline_vlg_tst/IR_IF	0045202b
/pipeline_vlg_tst/ALUOp_ID	0
/pipeline_vlg_tst/ALU_OpA	00000000
/pipeline_vlg_tst/ALU_OpB	00000000
/pipeline_vlg_tst/ALU_out	00000000
/pipeline_vlg_tst/ShiftAmount	00
/pipeline_vlg_tst/Shifter_out	00000000
/pipeline_vlg_tst/Data_out	xxxxxxxx
/pipeline_vlg_tst/PC_in	0000001c
/pipeline_vlg_tst/PC_out	00000018
/pipeline_vlg_tst/Jump_ID	0
/pipeline_vlg_tst/Rs_EX_Forward	0
/pipeline_vlg_tst/Rs_ID	00
/pipeline_vlg_tst/Rt_EX_Forward	0
/pipeline_vlg_tst/Rt_ID	00
/pipeline_vlg_tst/Rd_EX	00
/pipeline_vlg_tst/Rd_ID	00
/pipeline_vlg_tst/Rd_Mem	00
/pipeline_vlg_tst/Rd_Write_Byte...	0
/pipeline_vlg_tst/Rd_in	00000000
/pipeline_vlg_tst/WBData_EX	00000000
/pipeline_vlg_tst/MemData_EX	00000000
/pipeline_vlg_tst/MemWriteEn	0
/pipeline_vlg_tst/flash_EX	0
/pipeline_vlg_tst/flash_ID	0
/pipeline_vlg_tst/Immediate32_ID	00000000
/pipeline_vlg_tst/Immediate32_IF	X0XXX0XX
/pipeline_vlg_tst/RegShift	00000000
Now	306,800 ps



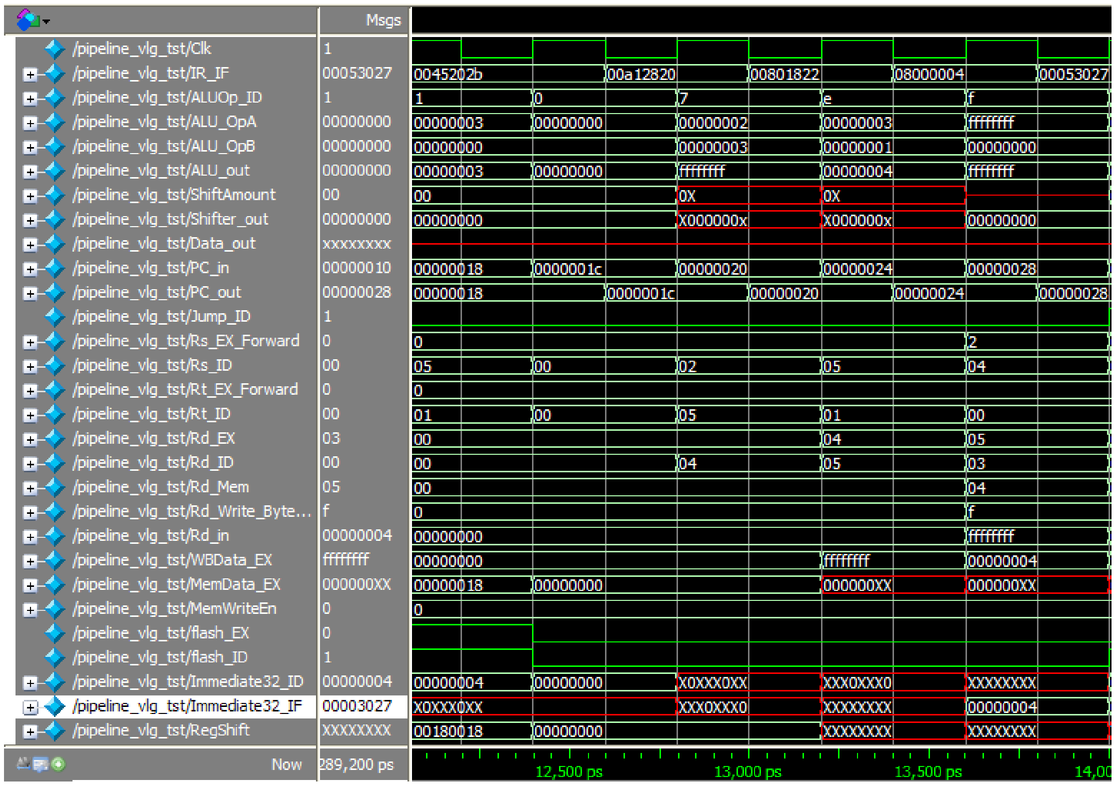
	Msgs
/pipeline_vlg_tst/Clk	1
/pipeline_vlg_tst/IR_IF	04610001
/pipeline_vlg_tst/ALUOp_ID	0
/pipeline_vlg_tst/ALU_OpA	00000000
/pipeline_vlg_tst/ALU_OpB	00000000
/pipeline_vlg_tst/ALU_out	00000000
/pipeline_vlg_tst/ShiftAmount	00
/pipeline_vlg_tst/Shifter_out	00000000
/pipeline_vlg_tst/Data_out	xxxxxxxx
/pipeline_vlg_tst/PC_in	00000014
/pipeline_vlg_tst/PC_out	00000010
/pipeline_vlg_tst/Jump_ID	0
/pipeline_vlg_tst/Rs_EX_Forward	0
/pipeline_vlg_tst/Rs_ID	00
/pipeline_vlg_tst/Rt_EX_Forward	0
/pipeline_vlg_tst/Rt_ID	00
/pipeline_vlg_tst/Rd_EX	00
/pipeline_vlg_tst/Rd_ID	00
/pipeline_vlg_tst/Rd_Mem	03
/pipeline_vlg_tst/Rd_Write_Byte...	f
/pipeline_vlg_tst/Rd_in	00000000
/pipeline_vlg_tst/WBData_EX	00000000
/pipeline_vlg_tst/MemData_EX	000000XX
/pipeline_vlg_tst/MemWriteEn	0
/pipeline_vlg_tst/flash_EX	0
/pipeline_vlg_tst/flash_ID	0
/pipeline_vlg_tst/Immediate32_ID	00000000
/pipeline_vlg_tst/Immediate32_IF	00000001
/pipeline_vlg_tst/RegShift	000000XX
Now	289,200 ps



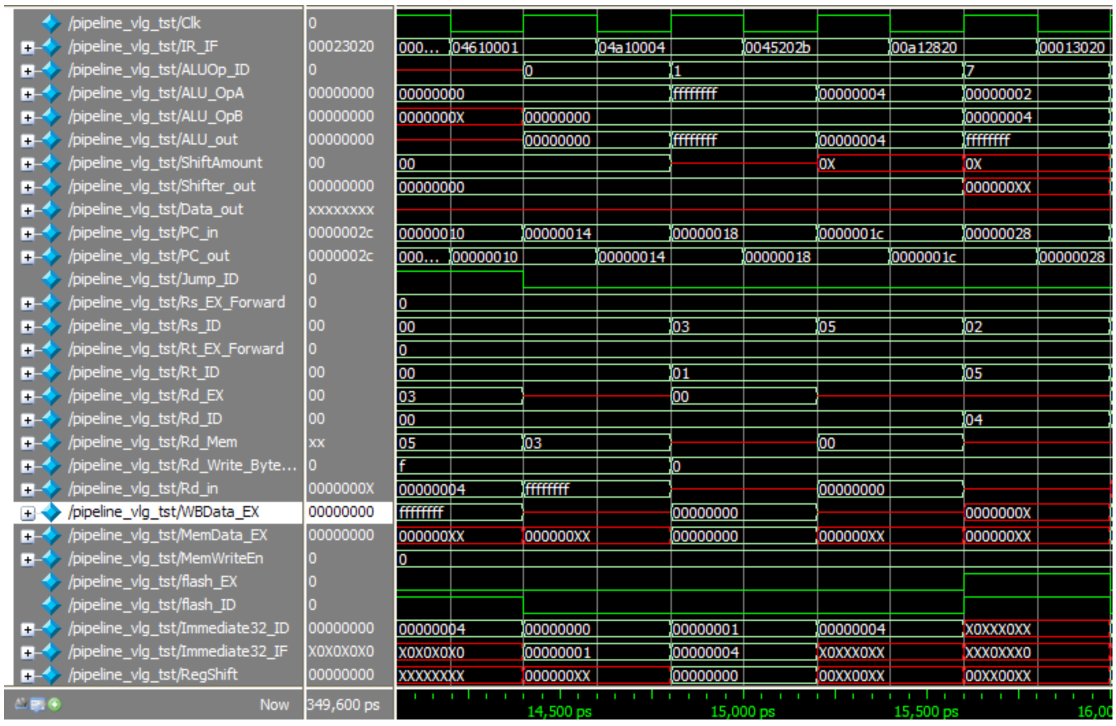
/pipeline_vlg_tst/Clk	1
/pipeline_vlg_tst/IR_IF	00801822
/pipeline_vlg_tst/ALUOp_ID	e
/pipeline_vlg_tst/ALU_OpA	00000002
/pipeline_vlg_tst/ALU_OpB	00000001
/pipeline_vlg_tst/ALU_out	00000003
/pipeline_vlg_tst/ShiftAmount	0X
/pipeline_vlg_tst/Shifter_out	X000000X
/pipeline_vlg_tst/Data_out	xxxxxxxx
/pipeline_vlg_tst/PC_in	00000024
/pipeline_vlg_tst/PC_out	00000020
/pipeline_vlg_tst/Jump_ID	0
/pipeline_vlg_tst/Rs_EX_Forward	0
/pipeline_vlg_tst/Rs_ID	05
/pipeline_vlg_tst/Rt_EX_Forward	0
/pipeline_vlg_tst/Rt_ID	01
/pipeline_vlg_tst/Rd_EX	04
/pipeline_vlg_tst/Rd_ID	05
/pipeline_vlg_tst/Rd_Mem	00
/pipeline_vlg_tst/Rd_Write_Byte...	0
/pipeline_vlg_tst/Rd_in	00000000
/pipeline_vlg_tst/WBData_EX	00000000
/pipeline_vlg_tst/MemData_EX	000000XX
/pipeline_vlg_tst/MemWriteEn	0
/pipeline_vlg_tst/flash_EX	0
/pipeline_vlg_tst/flash_ID	0
/pipeline_vlg_tst/Immediate32_ID	XXX0XXX0
/pipeline_vlg_tst/Immediate32_IF	XXXXXXXX
/pipeline_vlg_tst/RegShift	XXXXXXXX
Now	289,200 ps



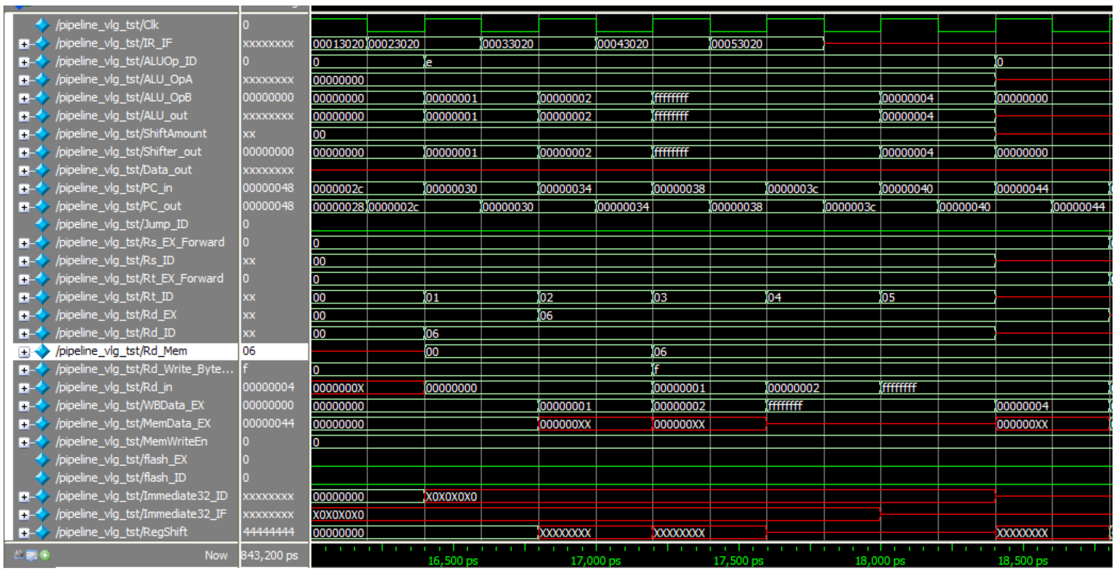
	Msgs
/pipeline_vlg_tst/Clk	1
/pipeline_vlg_tst/IR_IF	0045202b
/pipeline_vlg_tst/ALUOp_ID	1
/pipeline_vlg_tst/ALU_OpA	00000003
/pipeline_vlg_tst/ALU_OpB	00000000
/pipeline_vlg_tst/ALU_out	00000003
/pipeline_vlg_tst/ShiftAmount	00
/pipeline_vlg_tst/Shifter_out	00000000
/pipeline_vlg_tst/Data_out	xxxxxxxx
/pipeline_vlg_tst/PC_in	00000018
/pipeline_vlg_tst/PC_out	00000018
/pipeline_vlg_tst/Jump_ID	0
/pipeline_vlg_tst/Rs_EX_Forward	0
/pipeline_vlg_tst/Rs_ID	05
/pipeline_vlg_tst/Rt_EX_Forward	0
/pipeline_vlg_tst/Rt_ID	01
/pipeline_vlg_tst/Rd_EX	00
/pipeline_vlg_tst/Rd_ID	00
/pipeline_vlg_tst/Rd_Mem	00
/pipeline_vlg_tst/Rd_Write_Byte...	0
/pipeline_vlg_tst/Rd_in	00000000
/pipeline_vlg_tst/WBData_EX	00000000
/pipeline_vlg_tst/MemData_EX	00000018
/pipeline_vlg_tst/MemWriteEn	0
/pipeline_vlg_tst/flash_EX	1
/pipeline_vlg_tst/flash_ID	1
/pipeline_vlg_tst/Immediate32_ID	00000004
/pipeline_vlg_tst/Immediate32_IF	X0XXX0XX
/pipeline_vlg_tst/RegShift	00180018
Now	289,200 ps



	Msgs
/pipeline_vlg_tst/Clk	1
/pipeline_vlg_tst/IR_IF	00053027
/pipeline_vlg_tst/ALUOp_ID	1
/pipeline_vlg_tst/ALU_OpA	00000000
/pipeline_vlg_tst/ALU_OpB	00000000
/pipeline_vlg_tst/ALU_out	00000000
/pipeline_vlg_tst/ShiftAmount	00
/pipeline_vlg_tst/Shifter_out	00000000
/pipeline_vlg_tst/Data_out	xxxxxxxx
/pipeline_vlg_tst/PC_in	00000010
/pipeline_vlg_tst/PC_out	00000028
/pipeline_vlg_tst/Jump_ID	1
/pipeline_vlg_tst/Rs_EX_Forward	0
/pipeline_vlg_tst/Rs_ID	00
/pipeline_vlg_tst/Rt_EX_Forward	0
/pipeline_vlg_tst/Rt_ID	00
/pipeline_vlg_tst/Rd_EX	03
/pipeline_vlg_tst/Rd_ID	00
/pipeline_vlg_tst/Rd_Mem	05
/pipeline_vlg_tst/Rd_Write_Byte…	f
/pipeline_vlg_tst/Rd_in	00000004
/pipeline_vlg_tst/WBData_EX	ffffffff
/pipeline_vlg_tst/MemData_EX	000000XX
/pipeline_vlg_tst/MemWriteEn	0
/pipeline_vlg_tst/flash_EX	0
/pipeline_vlg_tst/flash_ID	1
/pipeline_vlg_tst/Immediate32_ID	00000004
/pipeline_vlg_tst/Immediate32_IF	00003027
/pipeline_vlg_tst/RegShift	XXXXXXXX
Now	289,200 ps



/pipeline_vlg_tst/Clk	0
/pipeline_vlg_tst/IR_IF	00023020
/pipeline_vlg_tst/ALUOp_ID	0
/pipeline_vlg_tst/ALU_OpA	00000000
/pipeline_vlg_tst/ALU_OpB	00000000
/pipeline_vlg_tst/ALU_out	00000000
/pipeline_vlg_tst/ShiftAmount	00
/pipeline_vlg_tst/Shifter_out	00000000
/pipeline_vlg_tst/Data_out	xxxxxxxx
/pipeline_vlg_tst/PC_in	0000002c
/pipeline_vlg_tst/PC_out	0000002c
/pipeline_vlg_tst/Jump_ID	0
/pipeline_vlg_tst/Rs_EX_Forward	0
/pipeline_vlg_tst/Rs_ID	00
/pipeline_vlg_tst/Rt_EX_Forward	0
/pipeline_vlg_tst/Rt_ID	00
/pipeline_vlg_tst/Rd_EX	00
/pipeline_vlg_tst/Rd_ID	00
/pipeline_vlg_tst/Rd_Mem	xx
/pipeline_vlg_tst/Rd_Write_Byte…	0
/pipeline_vlg_tst/Rd_in	0000000X
/pipeline_vlg_tst/WBData_EX	00000000
/pipeline_vlg_tst/MemData_EX	00000000
/pipeline_vlg_tst/MemWriteEn	0
/pipeline_vlg_tst/flash_EX	0
/pipeline_vlg_tst/flash_ID	0
/pipeline_vlg_tst/Immediate32_ID	00000000
/pipeline_vlg_tst/Immediate32_IF	X0X0X0X0
/pipeline_vlg_tst/RegShift	00000000
Now	349,600 ps


/pipeline_vlg_tst/Clk
/pipeline_vlg_tst/IR_IF
/pipeline_vlg_tst/ALUOp_ID
/pipeline_vlg_tst/ALU_OpA
/pipeline_vlg_tst/ALU_OpB
/pipeline_vlg_tst/ALU_out
/pipeline_vlg_tst/ShiftAmount
/pipeline_vlg_tst/Shifter_out
/pipeline_vlg_tst/Data_out
/pipeline_vlg_tst/PC_in
/pipeline_vlg_tst/PC_out
/pipeline_vlg_tst/Jump_ID
/pipeline_vlg_tst/Rs_EX_Forward
/pipeline_vlg_tst/Rs_ID
/pipeline_vlg_tst/Rt_EX_Forward
/pipeline_vlg_tst/Rt_ID
/pipeline_vlg_tst/Rd_EX
/pipeline_vlg_tst/Rd_ID
/pipeline_vlg_tst/Rd_Mem
/pipeline_vlg_tst/Rd_Write_Byte...
/pipeline_vlg_tst/Rd_in
/pipeline_vlg_tst/WBData_EX
/pipeline_vlg_tst/MemData_EX
/pipeline_vlg_tst/MemWriteEn
/pipeline_vlg_tst/flash_EX
/pipeline_vlg_tst/flash_ID
/pipeline_vlg_tst/Immediate32_ID
/pipeline_vlg_tst/Immediate32_IF
/pipeline_vlg_tst/RegShift
Now 843,200 ps
16,500 ps
17,000 ps
17,500 ps
18,000 ps
18,500 ps

conclusion:

图中可看出，最后的五个 add 得到的寄存器 r1~r5 的值都与预期值相同，连续执行跳转指令以及数据冒险和控制冒险同时发生的情况，该流水线 CPU 都能很好的解决，并无差错。

test-3:

本次测试在 test-2 的基础上加入了 sw，lw 指令，用于测试内存存取与其他类型指令冒险的结合性

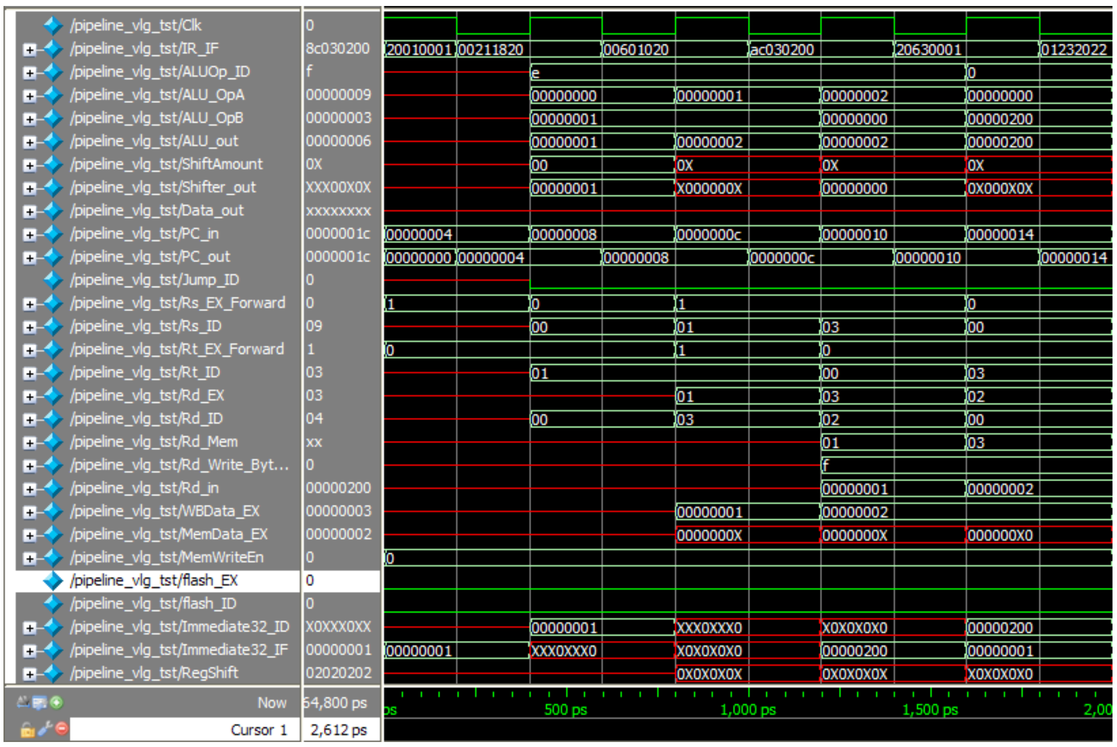
r10 为 9;

汇编代码的意义是：将 r1 加为 1，r2 加为 2，r3 加为 2，然后将 r3 存入内存的 512，然后进入循环，循环里做的是：r3 每回合+1，直到 r3 等于 r10，则 begz 不执行，就不跳过 lw，然后将 r3 从内存 512 中取出前面存入的 2，则执行 beg，跳过 jump，结束循环。

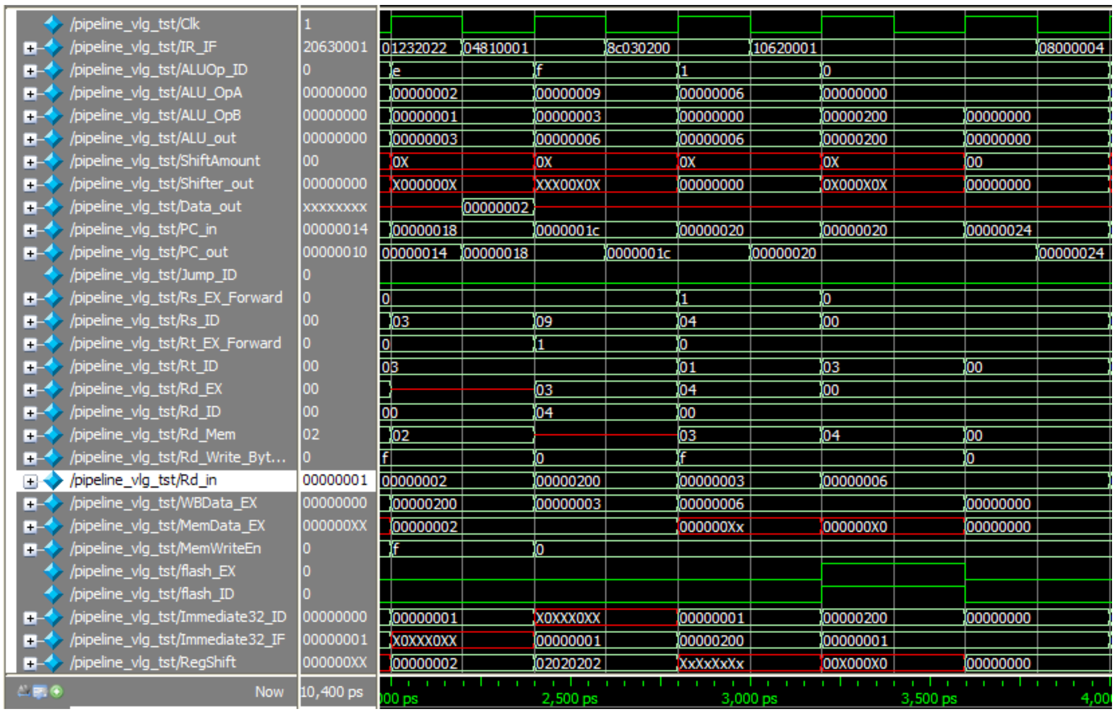
```
ram[0] = 32'b001000_00000_00001_0000000000000001;   //addi   r0 im(1) -> r1
ram[1] = 32'b000000_00001_00001_0001100000100000;   //add     r1 r1 -> r3
ram[2] = 32'b000000_00011_00000_0001000000100000;   //add     r3 r0 -> r2
ram[3] = {16'b101011_00000_00011,16'd512};           // sw    reg[3] -> DM[512]
ram[4] = 32'b001000_00011_00011_0000000000000001;   //addi   r3 im(1) -> r3
ram[5] = 32'b000000_01001_00011_0010000000100010;   //sub    r10 r3 ->r4
ram[6] = 32'b000001_00100_00001_0000000000000001;   //begz   r4 -> 1
ram[7] = {16'b100011_00000_00011,16'd512};           //lw    DM[512] -> reg[3]
ram[8] = 32'b000100_00011_00010_0000000000000001;   //beq    (r3 == r2)? ->1
ram[9] = 32'b000010_00000_00000_0000000000000100;   //j    to    4 (*4)
```

仿真截图：

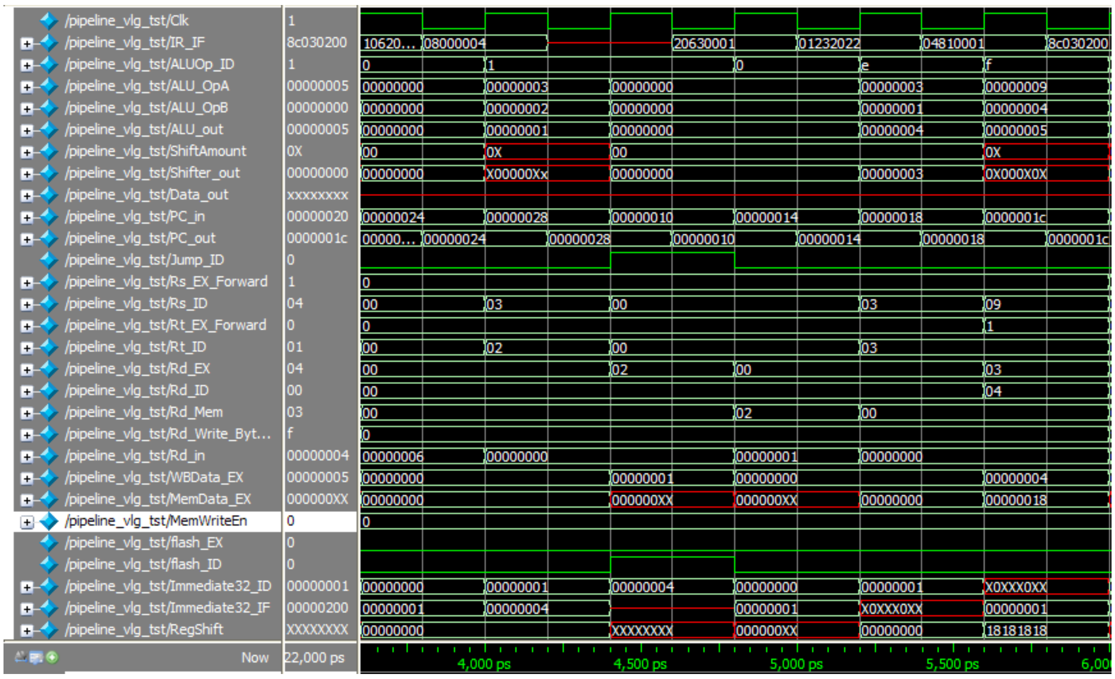
一下截图每张图五个周期，每张图五重复周期



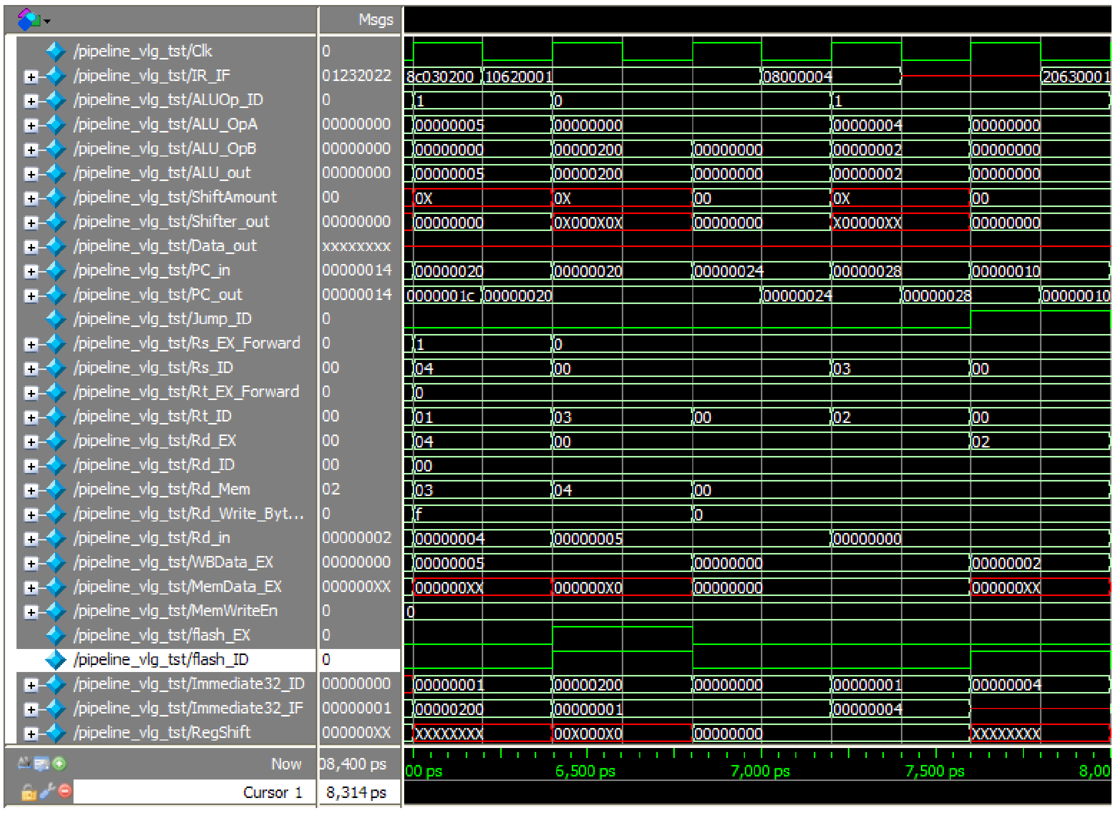
/pipeline_vlg_tst/Clk	0
/pipeline_vlg_tst/IR_IF	8c030200
/pipeline_vlg_tst/ALUOp_ID	f
/pipeline_vlg_tst/ALU_OpA	00000009
/pipeline_vlg_tst/ALU_OpB	00000003
/pipeline_vlg_tst/ALU_out	00000006
/pipeline_vlg_tst/ShiftAmount	0X
/pipeline_vlg_tst/Shifter_out	XXX00X0X
/pipeline_vlg_tst/Data_out	xxxxxxxx
/pipeline_vlg_tst/PC_in	0000001c
/pipeline_vlg_tst/PC_out	0000001c
/pipeline_vlg_tst/Jump_ID	0
/pipeline_vlg_tst/Rs_EX_Forward	0
/pipeline_vlg_tst/Rs_ID	09
/pipeline_vlg_tst/Rt_EX_Forward	1
/pipeline_vlg_tst/Rt_ID	03
/pipeline_vlg_tst/Rd_EX	03
/pipeline_vlg_tst/Rd_ID	04
/pipeline_vlg_tst/Rd_Mem	xx
/pipeline_vlg_tst/Rd_Write_Byt...	0
/pipeline_vlg_tst/Rd_in	00000200
/pipeline_vlg_tst/WBData_EX	00000003
/pipeline_vlg_tst/MemData_EX	00000002
/pipeline_vlg_tst/MemWriteEn	0
/pipeline_vlg_tst/flash_EX	0
/pipeline_vlg_tst/flash_ID	0
/pipeline_vlg_tst/Immediate32_ID	X0XXX0XX
/pipeline_vlg_tst/Immediate32_IF	00000001
/pipeline_vlg_tst/RegShift	02020202
Now	64,800 ps
Cursor 1	2,612 ps



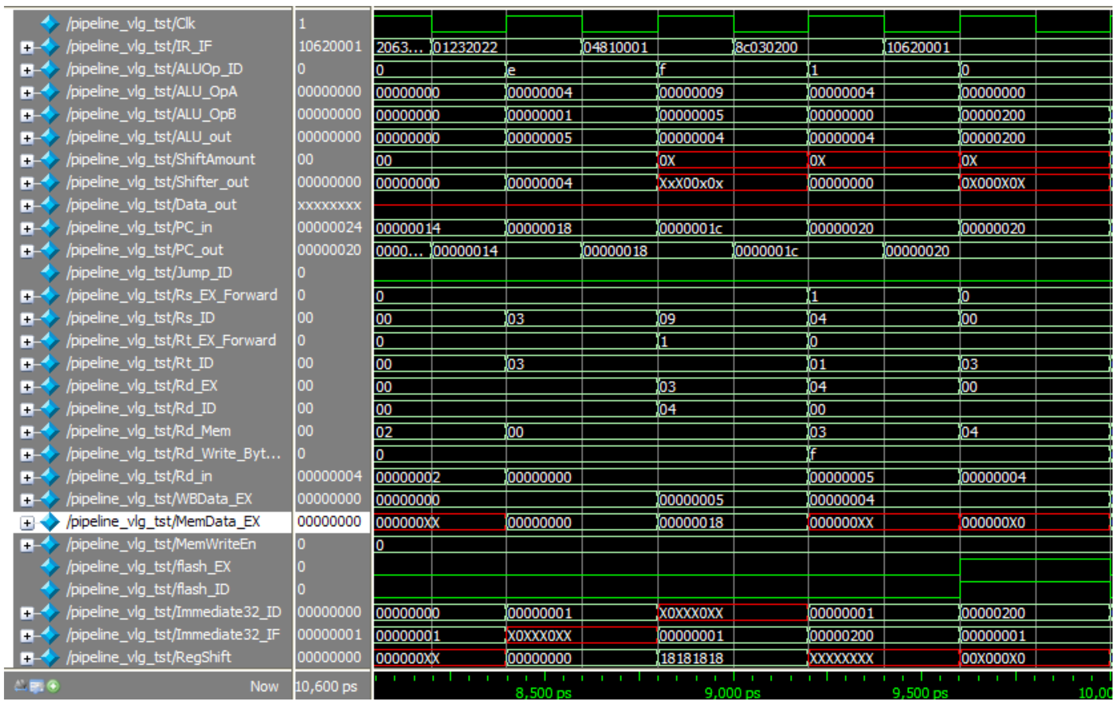
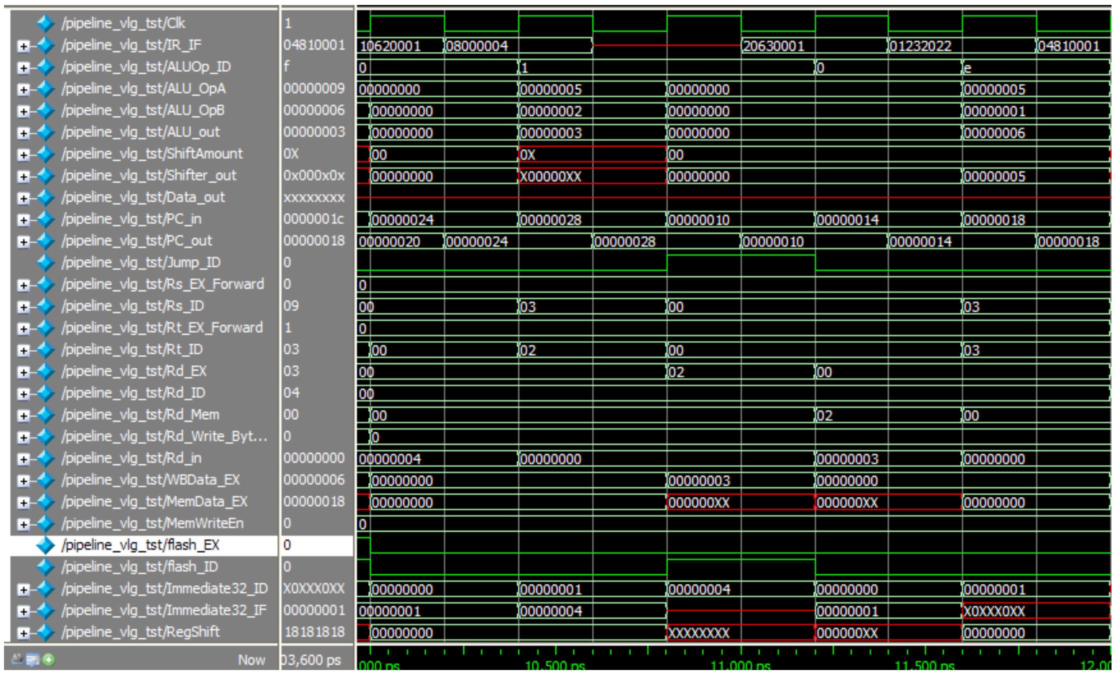
/pipeline_vlg_tst/Clk	1
/pipeline_vlg_tst/IR_IF	20630001
/pipeline_vlg_tst/ALUOp_ID	0
/pipeline_vlg_tst/ALU_OpA	00000000
/pipeline_vlg_tst/ALU_OpB	00000000
/pipeline_vlg_tst/ALU_out	00000000
/pipeline_vlg_tst/ShiftAmount	00
/pipeline_vlg_tst/Shifter_out	00000000
/pipeline_vlg_tst/Data_out	xxxxxxxx
/pipeline_vlg_tst/PC_in	00000014
/pipeline_vlg_tst/PC_out	00000010
/pipeline_vlg_tst/Jump_ID	0
/pipeline_vlg_tst/Rs_EX_Forward	0
/pipeline_vlg_tst/Rs_ID	00
/pipeline_vlg_tst/Rt_EX_Forward	0
/pipeline_vlg_tst/Rt_ID	00
/pipeline_vlg_tst/Rd_EX	00
/pipeline_vlg_tst/Rd_ID	00
/pipeline_vlg_tst/Rd_Mem	02
/pipeline_vlg_tst/Rd_Write_Byt...	0
/pipeline_vlg_tst/Rd_in	00000001
/pipeline_vlg_tst/WBData_EX	00000000
/pipeline_vlg_tst/MemData_EX	000000XX
/pipeline_vlg_tst/MemWriteEn	0
/pipeline_vlg_tst/flash_EX	0
/pipeline_vlg_tst/flash_ID	0
/pipeline_vlg_tst/Immediate32_ID	00000000
/pipeline_vlg_tst/Immediate32_IF	00000001
/pipeline_vlg_tst/RegShift	000000XX
Now	10,400 ps

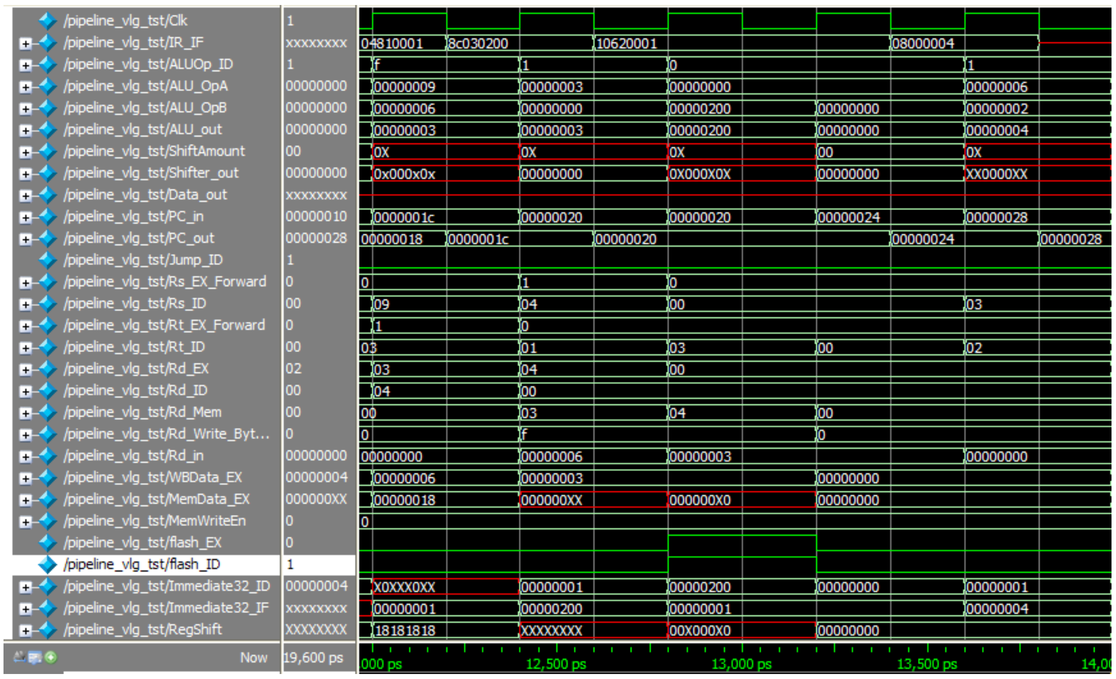


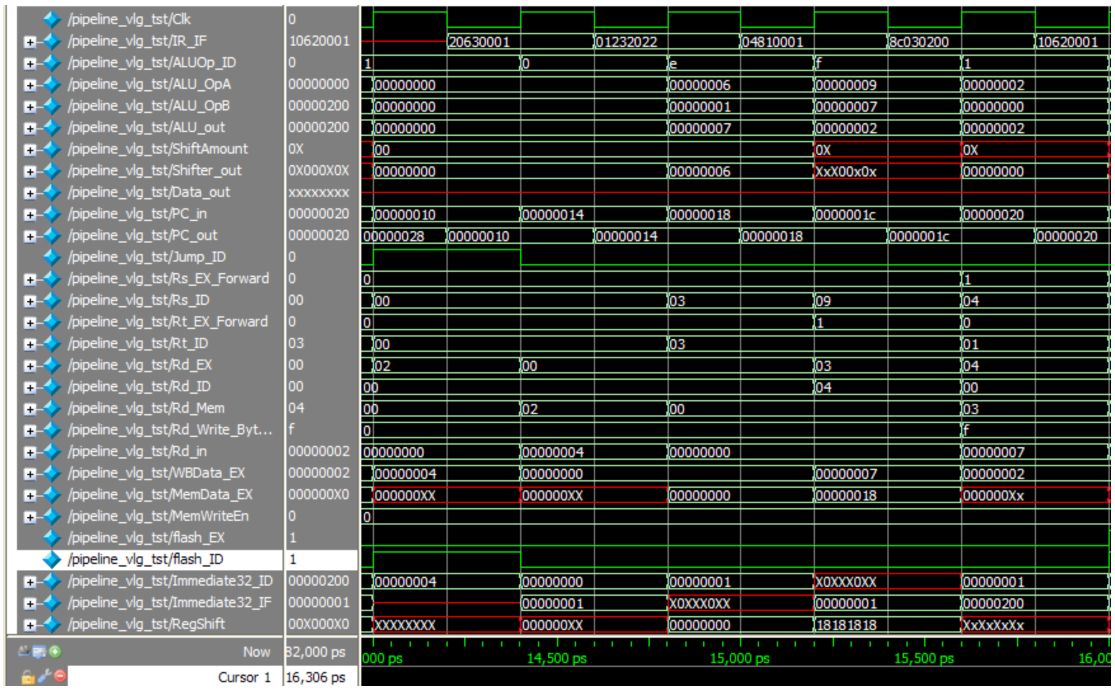
```
/pipeline_vlg_tst/Clk                1
/pipeline_vlg_tst/IR_IF              8c030200
/pipeline_vlg_tst/ALUOp_ID           1
/pipeline_vlg_tst/ALU_OpA            00000005
/pipeline_vlg_tst/ALU_OpB            00000000
/pipeline_vlg_tst/ALU_out            00000005
/pipeline_vlg_tst/ShiftAmount        0X
/pipeline_vlg_tst/Shifter_out        00000000
/pipeline_vlg_tst/Data_out           xxxxxxxx
/pipeline_vlg_tst/PC_in              00000020
/pipeline_vlg_tst/PC_out             0000001c
/pipeline_vlg_tst/Jump_ID            0
/pipeline_vlg_tst/Rs_EX_Forward      1
/pipeline_vlg_tst/Rs_ID              04
/pipeline_vlg_tst/Rt_EX_Forward      0
/pipeline_vlg_tst/Rt_ID              01
/pipeline_vlg_tst/Rd_EX              04
/pipeline_vlg_tst/Rd_ID              00
/pipeline_vlg_tst/Rd_Mem             03
/pipeline_vlg_tst/Rd_Write_Byt...    f
/pipeline_vlg_tst/Rd_in              00000004
/pipeline_vlg_tst/WBData_EX          00000005
/pipeline_vlg_tst/MemData_EX         000000XX
/pipeline_vlg_tst/MemWriteEn         0
/pipeline_vlg_tst/flash_EX           0
/pipeline_vlg_tst/flash_ID           0
/pipeline_vlg_tst/Immediate32_ID     00000001
/pipeline_vlg_tst/Immediate32_IF     00000200
/pipeline_vlg_tst/RegShift           XXXXXXXX
Now                                  22,000 ps
```

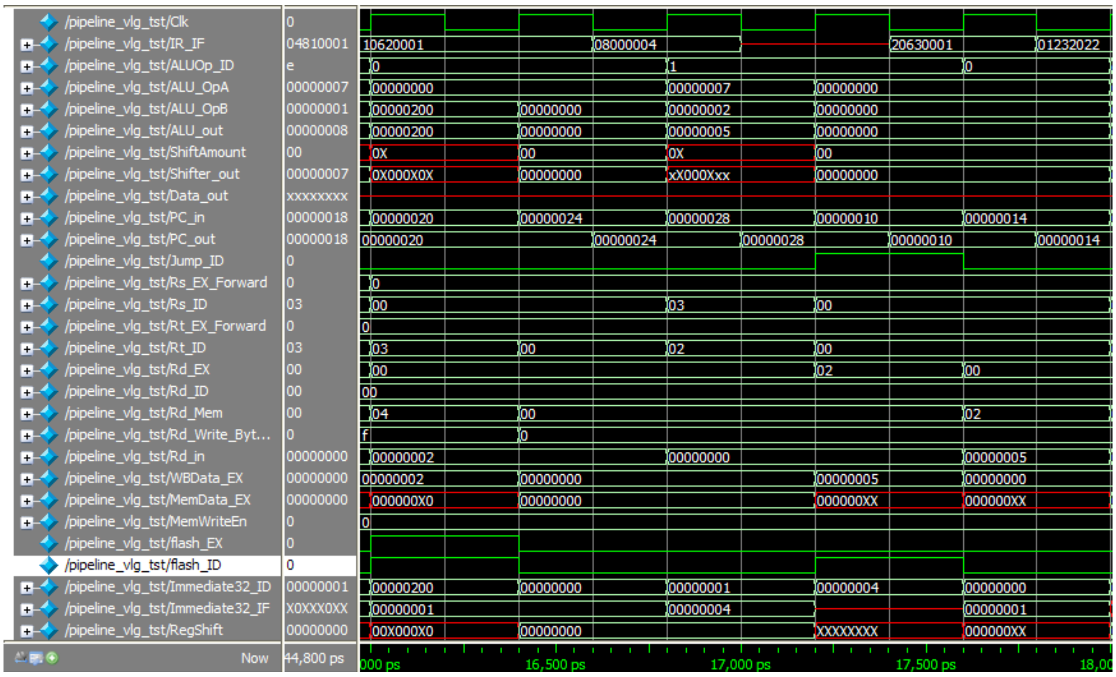


```
                                     Msgs
/pipeline_vlg_tst/Clk                0
/pipeline_vlg_tst/IR_IF              01232022
/pipeline_vlg_tst/ALUOp_ID           0
/pipeline_vlg_tst/ALU_OpA            00000000
/pipeline_vlg_tst/ALU_OpB            00000000
/pipeline_vlg_tst/ALU_out            00000000
/pipeline_vlg_tst/ShiftAmount        00
/pipeline_vlg_tst/Shifter_out        00000000
/pipeline_vlg_tst/Data_out           xxxxxxxx
/pipeline_vlg_tst/PC_in              00000014
/pipeline_vlg_tst/PC_out             00000014
/pipeline_vlg_tst/Jump_ID            0
/pipeline_vlg_tst/Rs_EX_Forward      0
/pipeline_vlg_tst/Rs_ID              00
/pipeline_vlg_tst/Rt_EX_Forward      0
/pipeline_vlg_tst/Rt_ID              00
/pipeline_vlg_tst/Rd_EX              00
/pipeline_vlg_tst/Rd_ID              00
/pipeline_vlg_tst/Rd_Mem             02
/pipeline_vlg_tst/Rd_Write_Byt...    0
/pipeline_vlg_tst/Rd_in              00000002
/pipeline_vlg_tst/WBData_EX          00000000
/pipeline_vlg_tst/MemData_EX         000000XX
/pipeline_vlg_tst/MemWriteEn         0
/pipeline_vlg_tst/flash_EX           0
/pipeline_vlg_tst/flash_ID           0
/pipeline_vlg_tst/Immediate32_ID     00000000
/pipeline_vlg_tst/Immediate32_IF     00000001
/pipeline_vlg_tst/RegShift           000000XX
Now                                  08,400 ps
Cursor 1                             8,314 ps
```

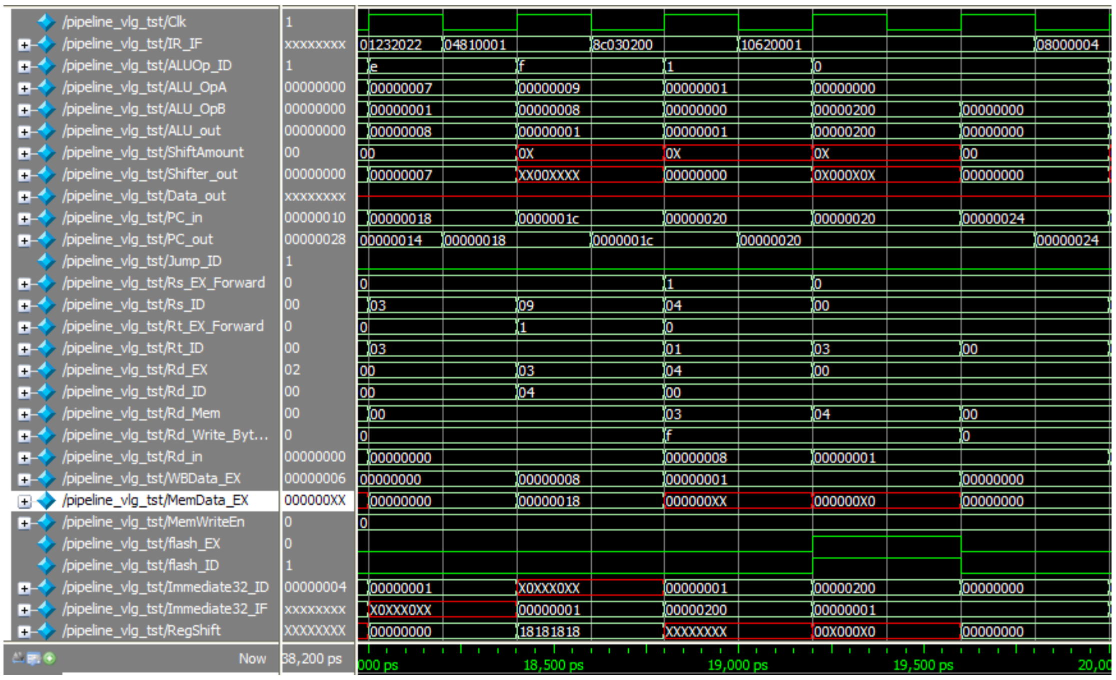

/pipeline_vlg_tst/Clk 1
/pipeline_vlg_tst/IR_IF 10620001
/pipeline_vlg_tst/ALUOp_ID 0
/pipeline_vlg_tst/ALU_OpA 00000000
/pipeline_vlg_tst/ALU_OpB 00000000
/pipeline_vlg_tst/ALU_out 00000000
/pipeline_vlg_tst/ShiftAmount 00
/pipeline_vlg_tst/Shifter_out 00000000
/pipeline_vlg_tst/Data_out xxxxxxxx
/pipeline_vlg_tst/PC_in 00000024
/pipeline_vlg_tst/PC_out 00000020
/pipeline_vlg_tst/Jump_ID 0
/pipeline_vlg_tst/Rs_EX_Forward 0
/pipeline_vlg_tst/Rs_ID 00
/pipeline_vlg_tst/Rt_EX_Forward 0
/pipeline_vlg_tst/Rt_ID 00
/pipeline_vlg_tst/Rd_EX 00
/pipeline_vlg_tst/Rd_ID 00
/pipeline_vlg_tst/Rd_Mem 00
/pipeline_vlg_tst/Rd_Write_Byt... 0
/pipeline_vlg_tst/Rd_in 00000004
/pipeline_vlg_tst/WBData_EX 00000000
/pipeline_vlg_tst/MemData_EX 00000000
/pipeline_vlg_tst/MemWriteEn 0
/pipeline_vlg_tst/flash_EX 0
/pipeline_vlg_tst/flash_ID 0
/pipeline_vlg_tst/Immediate32_ID 00000000
/pipeline_vlg_tst/Immediate32_IF 00000001
/pipeline_vlg_tst/RegShift 00000000
Now 10,600 ps
8,500 ps
9,000 ps
9,500 ps

/pipeline_vlg_tst/Clk 1
/pipeline_vlg_tst/IR_IF 04810001
/pipeline_vlg_tst/ALUOp_ID f
/pipeline_vlg_tst/ALU_OpA 00000009
/pipeline_vlg_tst/ALU_OpB 00000006
/pipeline_vlg_tst/ALU_out 00000003
/pipeline_vlg_tst/ShiftAmount 0X
/pipeline_vlg_tst/Shifter_out 0x000x0x
/pipeline_vlg_tst/Data_out xxxxxxxx
/pipeline_vlg_tst/PC_in 0000001c
/pipeline_vlg_tst/PC_out 00000018
/pipeline_vlg_tst/Jump_ID 0
/pipeline_vlg_tst/Rs_EX_Forward 0
/pipeline_vlg_tst/Rs_ID 09
/pipeline_vlg_tst/Rt_EX_Forward 1
/pipeline_vlg_tst/Rt_ID 03
/pipeline_vlg_tst/Rd_EX 03
/pipeline_vlg_tst/Rd_ID 04
/pipeline_vlg_tst/Rd_Mem 00
/pipeline_vlg_tst/Rd_Write_Byt... 0
/pipeline_vlg_tst/Rd_in 00000000
/pipeline_vlg_tst/WBData_EX 00000006
/pipeline_vlg_tst/MemData_EX 00000018
/pipeline_vlg_tst/MemWriteEn 0
/pipeline_vlg_tst/flash_EX 0
/pipeline_vlg_tst/flash_ID 0
/pipeline_vlg_tst/Immediate32_ID X0XXX0XX
/pipeline_vlg_tst/Immediate32_IF 00000001
/pipeline_vlg_tst/RegShift 18181818
Now 13,600 ps
10,500 ps
11,000 ps
11,500 ps


/pipeline_vlg_tst/Clk
/pipeline_vlg_tst/IR_IF
/pipeline_vlg_tst/ALUOp_ID
/pipeline_vlg_tst/ALU_OpA
/pipeline_vlg_tst/ALU_OpB
/pipeline_vlg_tst/ALU_out
/pipeline_vlg_tst/ShiftAmount
/pipeline_vlg_tst/Shifter_out
/pipeline_vlg_tst/Data_out
/pipeline_vlg_tst/PC_in
/pipeline_vlg_tst/PC_out
/pipeline_vlg_tst/Jump_ID
/pipeline_vlg_tst/Rs_EX_Forward
/pipeline_vlg_tst/Rs_ID
/pipeline_vlg_tst/Rt_EX_Forward
/pipeline_vlg_tst/Rt_ID
/pipeline_vlg_tst/Rd_EX
/pipeline_vlg_tst/Rd_ID
/pipeline_vlg_tst/Rd_Mem
/pipeline_vlg_tst/Rd_Write_Byt...
/pipeline_vlg_tst/Rd_in
/pipeline_vlg_tst/WBData_EX
/pipeline_vlg_tst/MemData_EX
/pipeline_vlg_tst/MemWriteEn
/pipeline_vlg_tst/flash_EX
/pipeline_vlg_tst/flash_ID
/pipeline_vlg_tst/Immediate32_ID
/pipeline_vlg_tst/Immediate32_IF
/pipeline_vlg_tst/RegShift
Now
19,600 ps
000 ps
12,500 ps
13,000 ps
13,500 ps

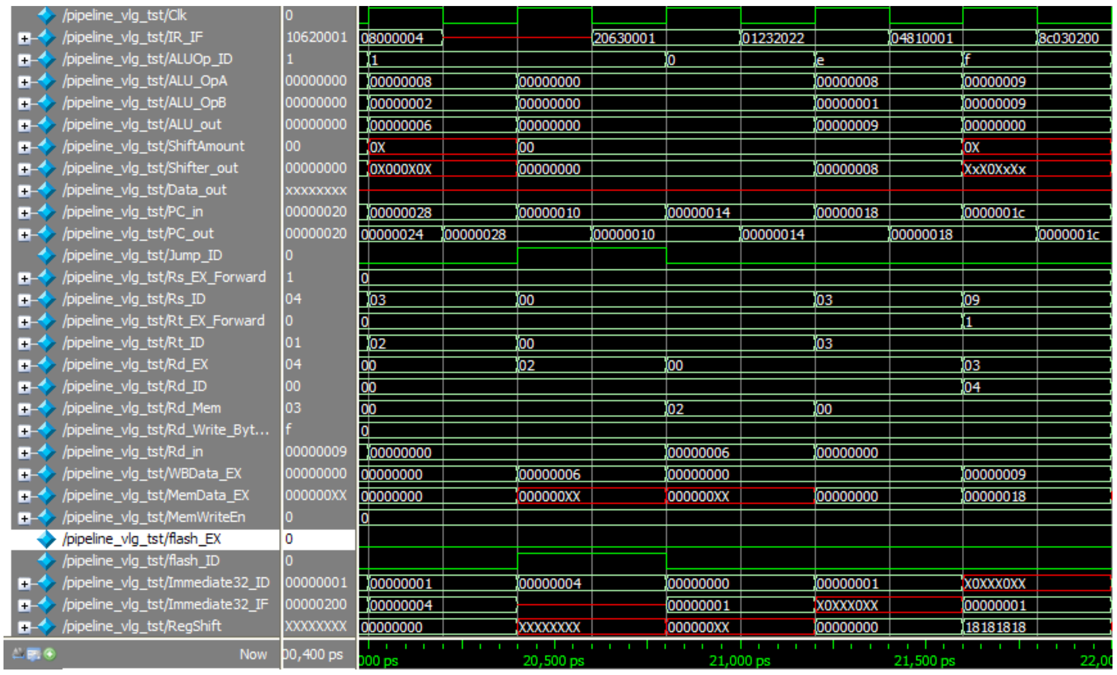

/pipeline_vlg_tst/Clk
/pipeline_vlg_tst/IR_IF
/pipeline_vlg_tst/ALUOp_ID
/pipeline_vlg_tst/ALU_OpA
/pipeline_vlg_tst/ALU_OpB
/pipeline_vlg_tst/ALU_out
/pipeline_vlg_tst/ShiftAmount
/pipeline_vlg_tst/Shifter_out
/pipeline_vlg_tst/Data_out
/pipeline_vlg_tst/PC_in
/pipeline_vlg_tst/PC_out
/pipeline_vlg_tst/Jump_ID
/pipeline_vlg_tst/Rs_EX_Forward
/pipeline_vlg_tst/Rs_ID
/pipeline_vlg_tst/Rt_EX_Forward
/pipeline_vlg_tst/Rt_ID
/pipeline_vlg_tst/Rd_EX
/pipeline_vlg_tst/Rd_ID
/pipeline_vlg_tst/Rd_Mem
/pipeline_vlg_tst/Rd_Write_Byt...
/pipeline_vlg_tst/Rd_in
/pipeline_vlg_tst/WBData_EX
/pipeline_vlg_tst/MemData_EX
/pipeline_vlg_tst/MemWriteEn
/pipeline_vlg_tst/flash_EX
/pipeline_vlg_tst/flash_ID
/pipeline_vlg_tst/Immediate32_ID
/pipeline_vlg_tst/Immediate32_IF
/pipeline_vlg_tst/RegShift
Now
32,000 ps
Cursor 1
16,306 ps
000 ps
14,500 ps
15,000 ps
15,500 ps

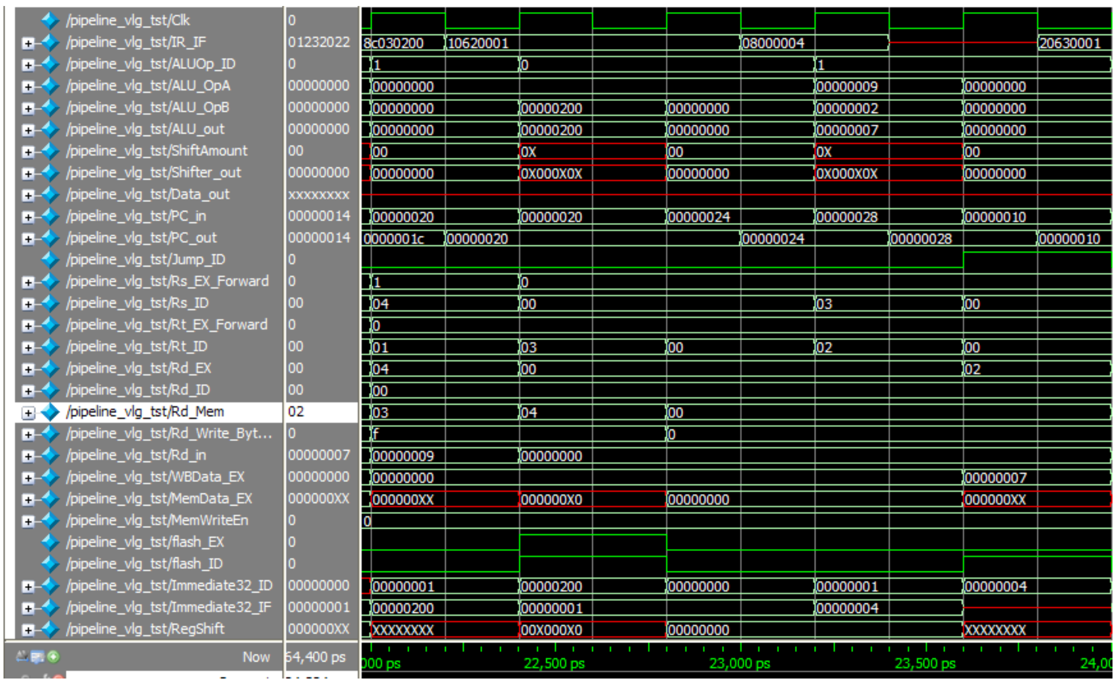


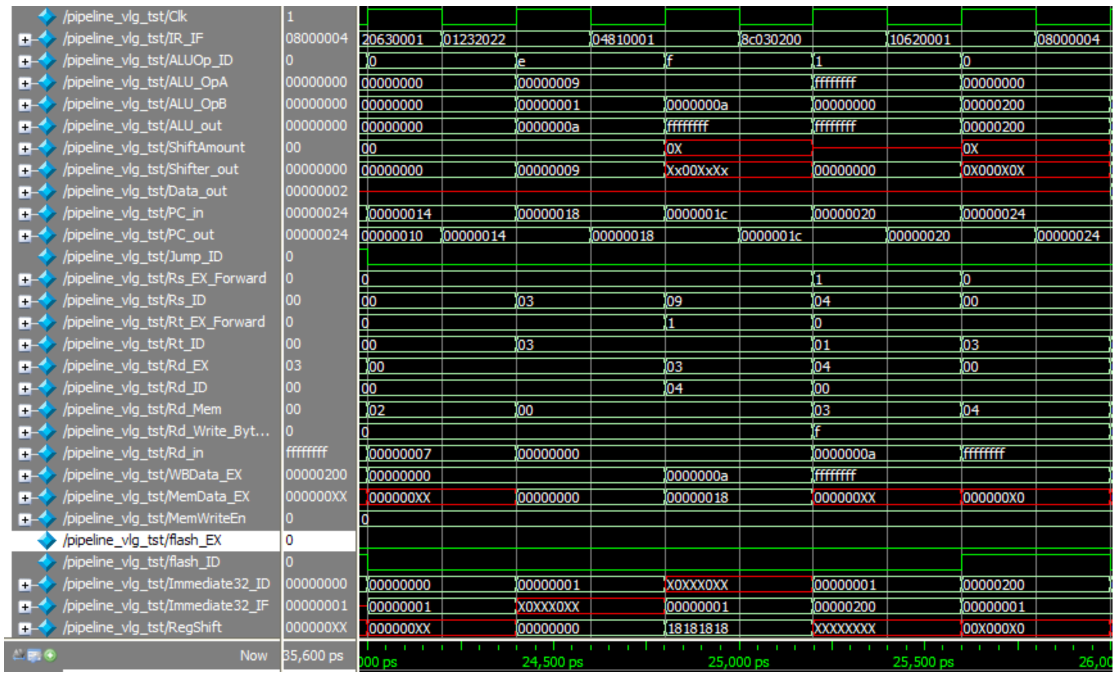
```
/pipeline_vlg_tst/Clk                0
/pipeline_vlg_tst/IR_IF              04810001
/pipeline_vlg_tst/ALUOp_ID           e
/pipeline_vlg_tst/ALU_OpA            00000007
/pipeline_vlg_tst/ALU_OpB            00000001
/pipeline_vlg_tst/ALU_out            00000008
/pipeline_vlg_tst/ShiftAmount        00
/pipeline_vlg_tst/Shifter_out        00000007
/pipeline_vlg_tst/Data_out           xxxxxxxx
/pipeline_vlg_tst/PC_in              00000018
/pipeline_vlg_tst/PC_out             00000018
/pipeline_vlg_tst/Jump_ID            0
/pipeline_vlg_tst/Rs_EX_Forward      0
/pipeline_vlg_tst/Rs_ID              03
/pipeline_vlg_tst/Rt_EX_Forward      0
/pipeline_vlg_tst/Rt_ID              03
/pipeline_vlg_tst/Rd_EX              00
/pipeline_vlg_tst/Rd_ID              00
/pipeline_vlg_tst/Rd_Mem             00
/pipeline_vlg_tst/Rd_Write_Byt...    0
/pipeline_vlg_tst/Rd_in              00000000
/pipeline_vlg_tst/WBData_EX          00000000
/pipeline_vlg_tst/MemData_EX         00000000
/pipeline_vlg_tst/MemWriteEn         0
/pipeline_vlg_tst/flash_EX           0
/pipeline_vlg_tst/flash_ID           0
/pipeline_vlg_tst/Immediate32_ID     00000001
/pipeline_vlg_tst/Immediate32_IF     X0XXX0XX
/pipeline_vlg_tst/RegShift           00000000
Now                                  44,800 ps
```

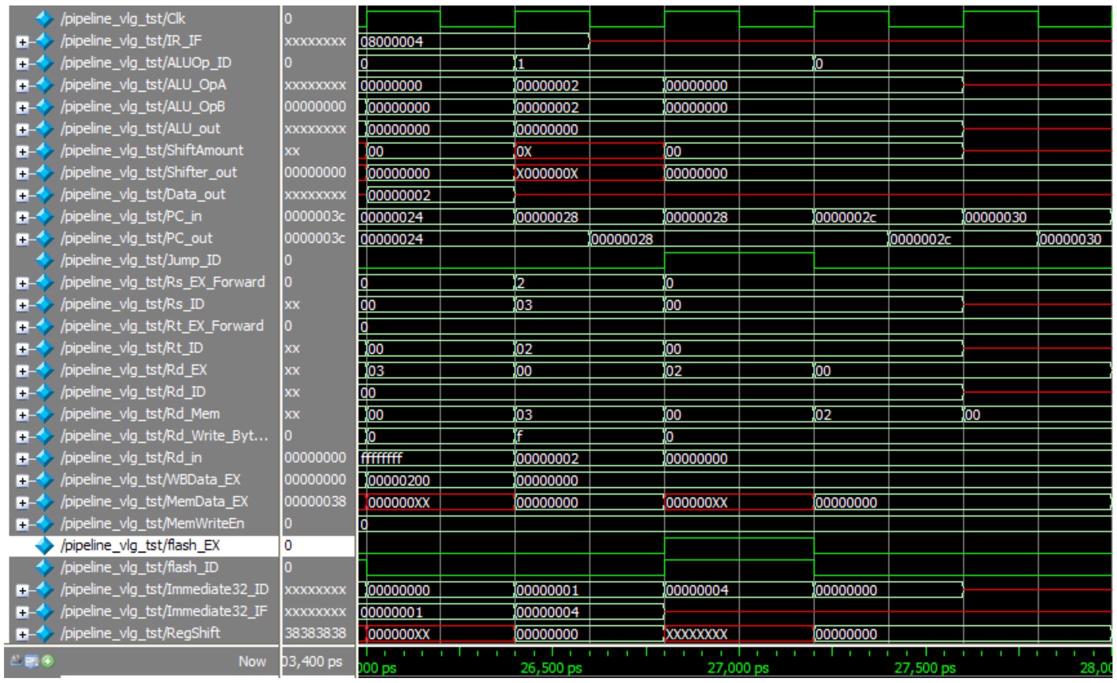


```
/pipeline_vlg_tst/Clk                1
/pipeline_vlg_tst/IR_IF              xxxxxxxx
/pipeline_vlg_tst/ALUOp_ID           1
/pipeline_vlg_tst/ALU_OpA            00000000
/pipeline_vlg_tst/ALU_OpB            00000000
/pipeline_vlg_tst/ALU_out            00000000
/pipeline_vlg_tst/ShiftAmount        00
/pipeline_vlg_tst/Shifter_out        00000000
/pipeline_vlg_tst/Data_out           xxxxxxxx
/pipeline_vlg_tst/PC_in              00000010
/pipeline_vlg_tst/PC_out             00000028
/pipeline_vlg_tst/Jump_ID            1
/pipeline_vlg_tst/Rs_EX_Forward      0
/pipeline_vlg_tst/Rs_ID              00
/pipeline_vlg_tst/Rt_EX_Forward      0
/pipeline_vlg_tst/Rt_ID              00
/pipeline_vlg_tst/Rd_EX              02
/pipeline_vlg_tst/Rd_ID              00
/pipeline_vlg_tst/Rd_Mem             00
/pipeline_vlg_tst/Rd_Write_Byt...    0
/pipeline_vlg_tst/Rd_in              00000000
/pipeline_vlg_tst/WBData_EX          00000006
/pipeline_vlg_tst/MemData_EX         000000XX
/pipeline_vlg_tst/MemWriteEn         0
/pipeline_vlg_tst/flash_EX           0
/pipeline_vlg_tst/flash_ID           1
/pipeline_vlg_tst/Immediate32_ID     00000004
/pipeline_vlg_tst/Immediate32_IF     xxxxxxxx
/pipeline_vlg_tst/RegShift           XXXXXXXX
Now                                  38,200 ps
```


/pipeline_vlg_tst/Clk 0
/pipeline_vlg_tst/IR_IF 10620001
/pipeline_vlg_tst/ALUOp_ID 1
/pipeline_vlg_tst/ALU_OpA 00000000
/pipeline_vlg_tst/ALU_OpB 00000000
/pipeline_vlg_tst/ALU_out 00000000
/pipeline_vlg_tst/ShiftAmount 00
/pipeline_vlg_tst/Shifter_out 00000000
/pipeline_vlg_tst/Data_out xxxxxxxx
/pipeline_vlg_tst/PC_in 00000020
/pipeline_vlg_tst/PC_out 00000020
/pipeline_vlg_tst/Jump_ID 0
/pipeline_vlg_tst/Rs_EX_Forward 1
/pipeline_vlg_tst/Rs_ID 04
/pipeline_vlg_tst/Rt_EX_Forward 0
/pipeline_vlg_tst/Rt_ID 01
/pipeline_vlg_tst/Rd_EX 04
/pipeline_vlg_tst/Rd_ID 00
/pipeline_vlg_tst/Rd_Mem 03
/pipeline_vlg_tst/Rd_Write_Byt... f
/pipeline_vlg_tst/Rd_in 00000009
/pipeline_vlg_tst/WBData_EX 00000000
/pipeline_vlg_tst/MemData_EX 000000XX
/pipeline_vlg_tst/MemWriteEn 0
/pipeline_vlg_tst/flash_EX 0
/pipeline_vlg_tst/flash_ID 0
/pipeline_vlg_tst/Immediate32_ID 00000001
/pipeline_vlg_tst/Immediate32_IF 00000200
/pipeline_vlg_tst/RegShift XXXXXXXX
Now 20,400 ps
000 ps
20,500 ps
21,000 ps
21,500 ps
22,00


/pipeline_vlg_tst/Clk 0
/pipeline_vlg_tst/IR_IF 01232022
/pipeline_vlg_tst/ALUOp_ID 0
/pipeline_vlg_tst/ALU_OpA 00000000
/pipeline_vlg_tst/ALU_OpB 00000000
/pipeline_vlg_tst/ALU_out 00000000
/pipeline_vlg_tst/ShiftAmount 00
/pipeline_vlg_tst/Shifter_out 00000000
/pipeline_vlg_tst/Data_out xxxxxxxx
/pipeline_vlg_tst/PC_in 00000014
/pipeline_vlg_tst/PC_out 00000014
/pipeline_vlg_tst/Jump_ID 0
/pipeline_vlg_tst/Rs_EX_Forward 0
/pipeline_vlg_tst/Rs_ID 00
/pipeline_vlg_tst/Rt_EX_Forward 0
/pipeline_vlg_tst/Rt_ID 00
/pipeline_vlg_tst/Rd_EX 00
/pipeline_vlg_tst/Rd_ID 00
/pipeline_vlg_tst/Rd_Mem 02
/pipeline_vlg_tst/Rd_Write_Byt... 0
/pipeline_vlg_tst/Rd_in 00000007
/pipeline_vlg_tst/WBData_EX 00000000
/pipeline_vlg_tst/MemData_EX 000000XX
/pipeline_vlg_tst/MemWriteEn 0
/pipeline_vlg_tst/flash_EX 0
/pipeline_vlg_tst/flash_ID 0
/pipeline_vlg_tst/Immediate32_ID 00000000
/pipeline_vlg_tst/Immediate32_IF 00000001
/pipeline_vlg_tst/RegShift 000000XX
Now 54,400 ps
000 ps
22,500 ps
23,000 ps
23,500 ps
24,00


/pipeline_vlg_tst/Clk
/pipeline_vlg_tst/IR_IF
/pipeline_vlg_tst/ALUOp_ID
/pipeline_vlg_tst/ALU_OpA
/pipeline_vlg_tst/ALU_OpB
/pipeline_vlg_tst/ALU_out
/pipeline_vlg_tst/ShiftAmount
/pipeline_vlg_tst/Shifter_out
/pipeline_vlg_tst/Data_out
/pipeline_vlg_tst/PC_in
/pipeline_vlg_tst/PC_out
/pipeline_vlg_tst/Jump_ID
/pipeline_vlg_tst/Rs_EX_Forward
/pipeline_vlg_tst/Rs_ID
/pipeline_vlg_tst/Rt_EX_Forward
/pipeline_vlg_tst/Rt_ID
/pipeline_vlg_tst/Rd_EX
/pipeline_vlg_tst/Rd_ID
/pipeline_vlg_tst/Rd_Mem
/pipeline_vlg_tst/Rd_Write_Byt...
/pipeline_vlg_tst/Rd_in
/pipeline_vlg_tst/WBData_EX
/pipeline_vlg_tst/MemData_EX
/pipeline_vlg_tst/MemWriteEn
/pipeline_vlg_tst/flash_EX
/pipeline_vlg_tst/flash_ID
/pipeline_vlg_tst/Immediate32_ID
/pipeline_vlg_tst/Immediate32_IF
/pipeline_vlg_tst/RegShift
Now
35,600 ps
24,500 ps
25,000 ps
25,500 ps


/pipeline_vlg_tst/Clk
/pipeline_vlg_tst/IR_IF
/pipeline_vlg_tst/ALUOp_ID
/pipeline_vlg_tst/ALU_OpA
/pipeline_vlg_tst/ALU_OpB
/pipeline_vlg_tst/ALU_out
/pipeline_vlg_tst/ShiftAmount
/pipeline_vlg_tst/Shifter_out
/pipeline_vlg_tst/Data_out
/pipeline_vlg_tst/PC_in
/pipeline_vlg_tst/PC_out
/pipeline_vlg_tst/Jump_ID
/pipeline_vlg_tst/Rs_EX_Forward
/pipeline_vlg_tst/Rs_ID
/pipeline_vlg_tst/Rt_EX_Forward
/pipeline_vlg_tst/Rt_ID
/pipeline_vlg_tst/Rd_EX
/pipeline_vlg_tst/Rd_ID
/pipeline_vlg_tst/Rd_Mem
/pipeline_vlg_tst/Rd_Write_Byt...
/pipeline_vlg_tst/Rd_in
/pipeline_vlg_tst/WBData_EX
/pipeline_vlg_tst/MemData_EX
/pipeline_vlg_tst/MemWriteEn
/pipeline_vlg_tst/flash_EX
/pipeline_vlg_tst/flash_ID
/pipeline_vlg_tst/Immediate32_ID
/pipeline_vlg_tst/Immediate32_IF
/pipeline_vlg_tst/RegShift
Now
03,400 ps
26,500 ps
27,000 ps
27,500 ps

conclusion:

图中可看出，连续执行跳转指令以及数据冒险和控制冒险同时发生的情况，该流水线CPU都能很好的解决，并无差错,在其中插入从内存中存取也没什么问题。

三、程序性检测：

计数器：

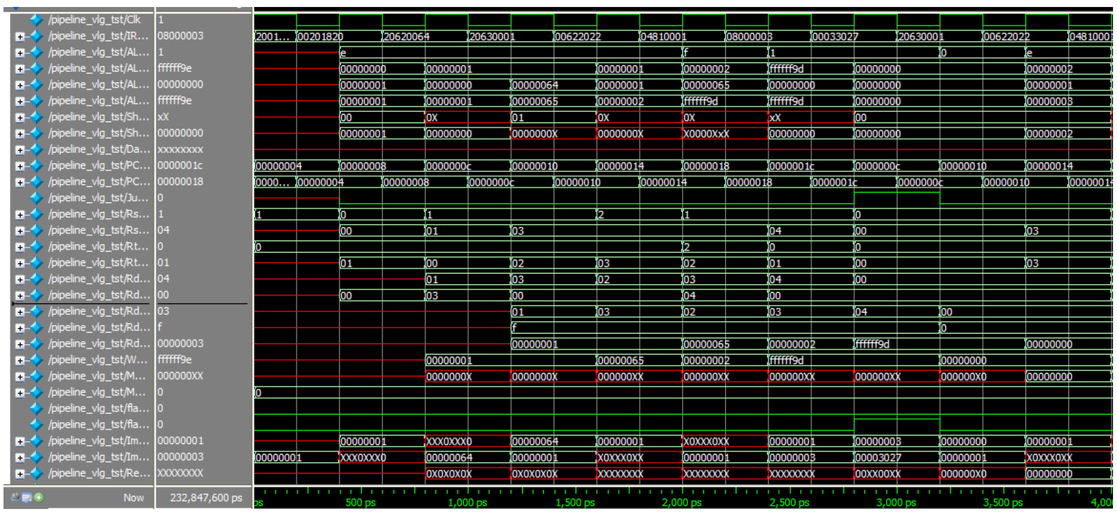
程序把 r3 从 1 加到 100

二进制代码：

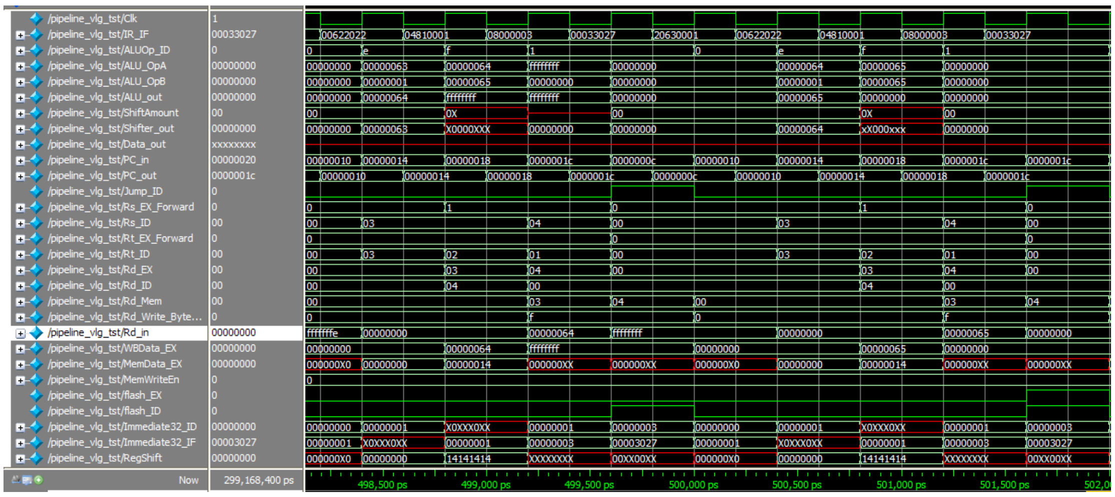
```
ram[0] = 32'b001000_00000_00001_0000000000000001;   //addi   r0 im(1) -> r1
ram[1] = 32'b000000_00001_00000_0001100000100000;   //add    r1 r0 -> r3
ram[2] = 32'b001000_00011_00010_0000000001100100;   //addi   r3 im(100) -> r2
ram[3] = 32'b001000_00011_00011_0000000000000001;   //addi   r3 im(1) -> r3
ram[4] = 32'b000000_00011_00010_0010000000100010;   //sub    r3 r2 ->r4
ram[5] = 32'b000001_00100_00001_0000000000000001;   //begz   r4 -> 1
ram[6] = 32'b000010_00000_00000_0000000000000011;   //j   to   3 (*4)
ram[7] = 32'b000000_00000_00011_00110_00000100111; //nor   r0 r3 -> r6
```

仿真截图：

为了提高效率，以下截图只截了程序开始十个周期和最后十个周期



/pipeline_vlg_tst/Clk 1
/pipeline_vlg_tst/IR... 08000003
/pipeline_vlg_tst/AL... 1
/pipeline_vlg_tst/AL... ffffff9e
/pipeline_vlg_tst/AL... 00000000
/pipeline_vlg_tst/AL... ffffff9e
/pipeline_vlg_tst/Sh... xX
/pipeline_vlg_tst/Sh... 00000000
/pipeline_vlg_tst/Da... xxxxxxxx
/pipeline_vlg_tst/PC... 0000001c
/pipeline_vlg_tst/PC... 00000018
/pipeline_vlg_tst/Ju... 0
/pipeline_vlg_tst/Rs... 1
/pipeline_vlg_tst/Rs... 04
/pipeline_vlg_tst/Rt... 0
/pipeline_vlg_tst/Rt... 01
/pipeline_vlg_tst/Rd... 04
/pipeline_vlg_tst/Rd... 00
/pipeline_vlg_tst/Rd... 03
/pipeline_vlg_tst/Rd... f
/pipeline_vlg_tst/Rd... 00000003
/pipeline_vlg_tst/W... ffffff9e
/pipeline_vlg_tst/M... 000000XX
/pipeline_vlg_tst/M... 0
/pipeline_vlg_tst/fla... 0
/pipeline_vlg_tst/fla... 0
/pipeline_vlg_tst/Im... 00000001
/pipeline_vlg_tst/Im... 00000003
/pipeline_vlg_tst/Re... XXXXXXXX
Now 232,847,600 ps
500 ps
1,000 ps
1,500 ps
2,000 ps
2,500 ps
3,000 ps
3,500 ps



```
/pipeline_vlg_tst/Clk                1
/pipeline_vlg_tst/IR_IF              00033027
/pipeline_vlg_tst/ALUOp_ID           0
/pipeline_vlg_tst/ALU_OpA            00000000
/pipeline_vlg_tst/ALU_OpB            00000000
/pipeline_vlg_tst/ALU_out            00000000
/pipeline_vlg_tst/ShiftAmount        00
/pipeline_vlg_tst/Shifter_out        00000000
/pipeline_vlg_tst/Data_out           xxxxxxxx
/pipeline_vlg_tst/PC_in              00000020
/pipeline_vlg_tst/PC_out             0000001c
/pipeline_vlg_tst/Jump_ID            0
/pipeline_vlg_tst/Rs_EX_Forward      0
/pipeline_vlg_tst/Rs_ID              00
/pipeline_vlg_tst/Rt_EX_Forward      0
/pipeline_vlg_tst/Rt_ID              00
/pipeline_vlg_tst/Rd_EX              00
/pipeline_vlg_tst/Rd_ID              00
/pipeline_vlg_tst/Rd_Mem             00
/pipeline_vlg_tst/Rd_Write_Byte...   0
/pipeline_vlg_tst/Rd_in              00000000
/pipeline_vlg_tst/WBData_EX          00000000
/pipeline_vlg_tst/MemData_EX         00000000
/pipeline_vlg_tst/MemWriteEn         0
/pipeline_vlg_tst/flash_EX           0
/pipeline_vlg_tst/flash_ID           0
/pipeline_vlg_tst/Immediate32_ID     00000000
/pipeline_vlg_tst/Immediate32_IF     00003027
/pipeline_vlg_tst/RegShift           00000000
Now                                  299,168,400 ps
```

冒泡排序：

寄存器 r1~r5 五个数据进行冒泡排序

初始寄存器的值为

```
register[0] = 0;
register[1] = 1;
register[2] = 5;
register[3] = 4;
register[4] = 3;
register[5] = 2;
```

然后进行冒泡排序

即进行两两比较，若后者小，就交换，反之不交换，如此进行四次。

若后者大，采用 begz 指令来跳过交换过程。最后采用 j 指令来实现循环执行，循环四次之后，采用一个 begz 指令过 j 指令，完成排序。

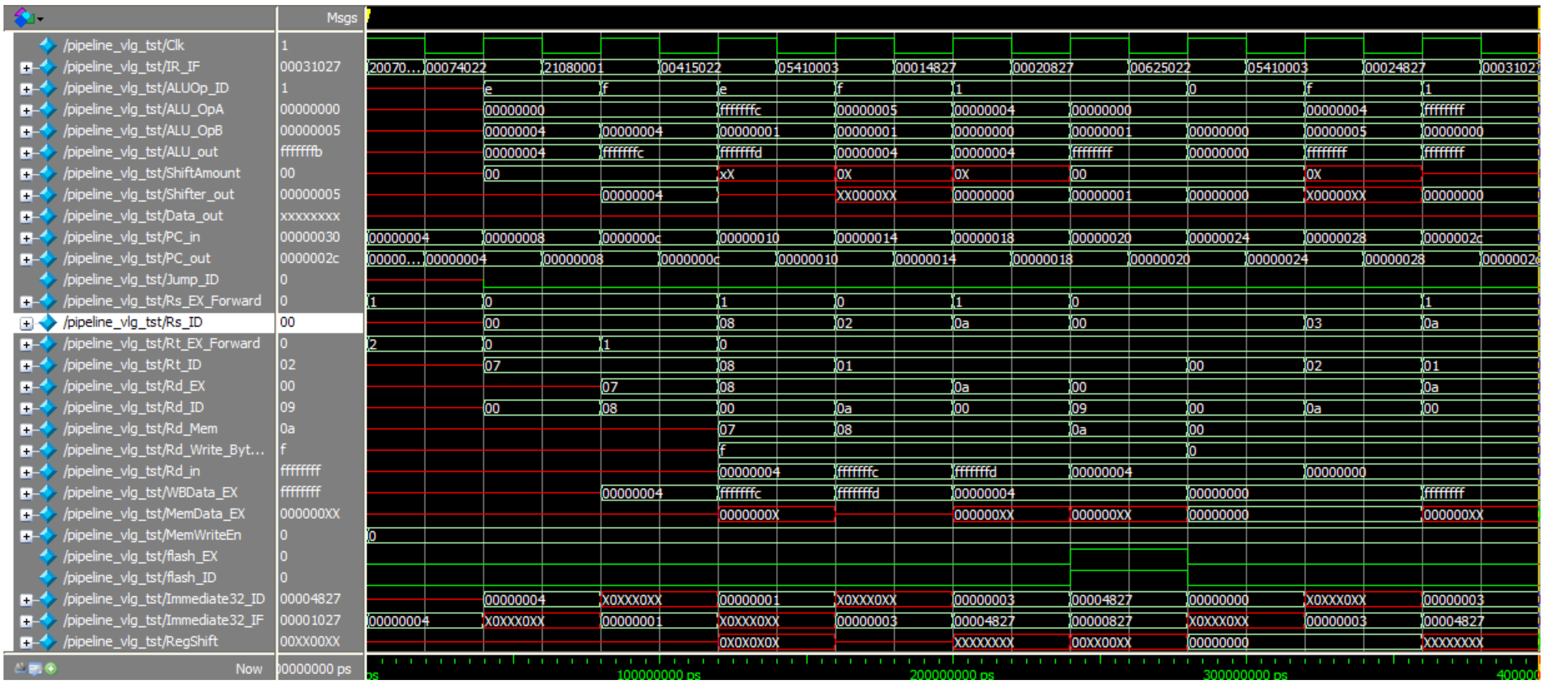
二进制代码：

```
ram[0] = 32'b001000_00000_00111_0000000000000100;    //addi r0 im(4) -> r7
ram[1] = 32'b000000_00000_00111_0100000000100010;    //sub   r0 r7 -> r8
ram[2] = 32'b001000_01000_01000_0000000000000001;    //addi r8 im(1) -> r8

ram[3] = 32'b000000_00010_00001_0101000000100010;    //sub   r2 r1 -> r10
ram[4] = 32'b000001_01010_00001_0000000000000011; //bgez r10 -> 3
ram[5] = 32'b000000_00000_00001_0100100000100000; //add r0 r1 -> r9
ram[6] = 32'b000000_00000_00010_0000100000100000; //add r0 r2 -> r1
ram[7] = 32'b000000_00000_01001_0001000000100000; //add r0 r9 -> r2

ram[8] = 32'b000000_00011_00010_0101000000100010;    //sub   r3 r2 -> r10
ram[9] = 32'b000001_01010_00001_0000000000000011; //bgez r10 -> 3
ram[10] = 32'b000000_00000_00010_0100100000100000; //add r0 r2 -> r9
ram[11] = 32'b000000_00000_00011_0001000000100000; //add r0 r3 -> r2
```

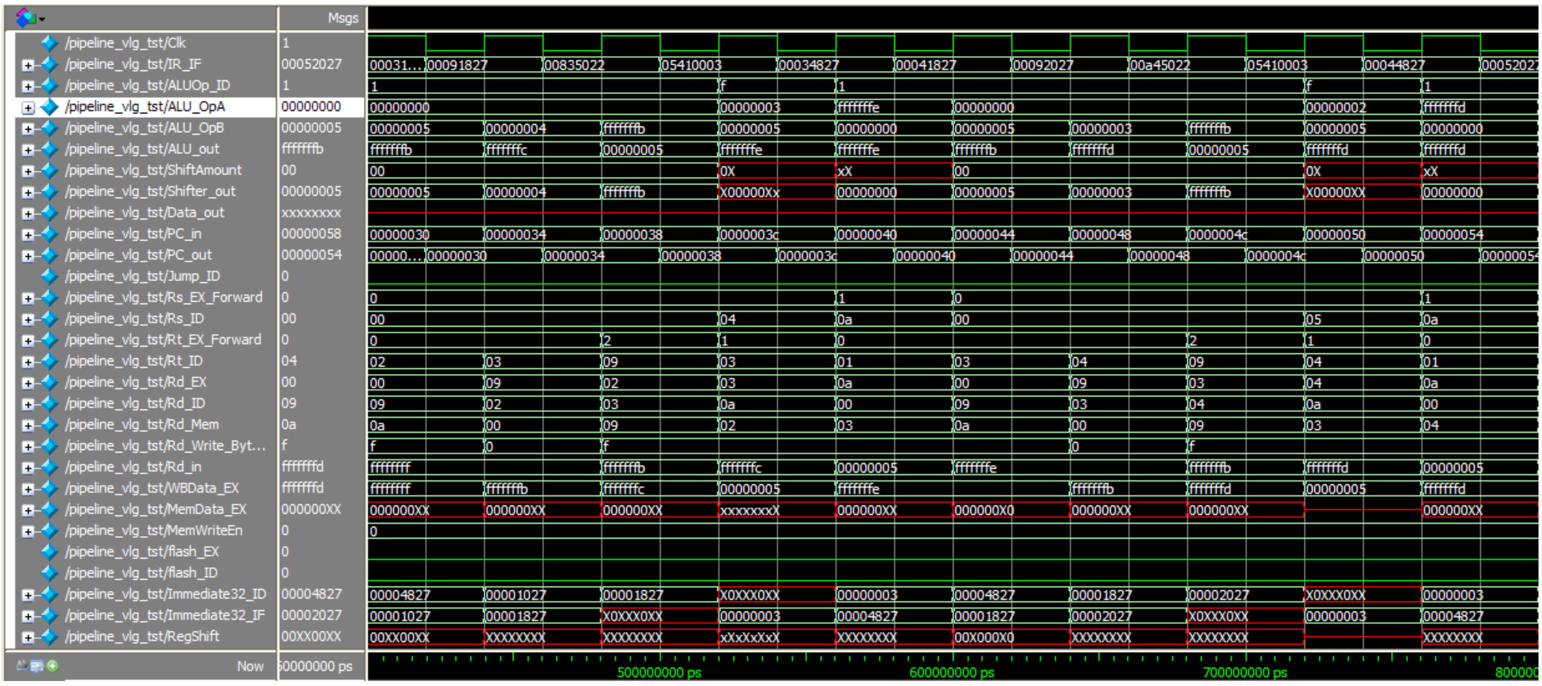
```
ram[12] = 32'b000000_00000_01001_0001100000100000; //add r0 r9 -> r3

ram[13] = 32'b000000_00100_00011_0101000000100010;    //sub    r4 r3 -> r10
ram[14] = 32'b000001_01010_00001_0000000000000011; //bgez r10 -> 3
ram[15] = 32'b000000_00000_00011_0100100000100000; //add r0 r3 -> r9
ram[16] = 32'b000000_00000_00100_0001100000100000; //add r0 r4 -> r3
ram[17] = 32'b000000_00000_01001_0010000000100000; //add r0 r9 -> r4

ram[18] = 32'b000000_00101_00100_0101000000100010;    //sub    r5 r4 -> r10
ram[19] = 32'b000001_01010_00001_0000000000000011; //bgez r10 -> 3
ram[20] = 32'b000000_00000_00100_0100100000100000; //add r0 r4 -> r9
ram[21] = 32'b000000_00000_00101_0010000000100000; //add r0 r5 -> r4
ram[22] = 32'b000000_00000_01001_0010100000100000; //add r0 r9 -> r5

ram[23] = 32'b000001_01000_00001_0000000000000001; //bgez r8 -> 1
ram[24] = 32'b000010_00000_00000_0000000000000010; //j       jump to 2
ram[25] = 32'b001000_00000_11110_0000000000001111;    //addi r0 im(15) -> r30
ram[26] = 32'b001000_00001_00111_0000000000000000;    //addi r1 im(0) -> r7
ram[27] = 32'b001000_00010_00111_0000000000000000;    //addi r2 im(0) -> r7
ram[28] = 32'b001000_00011_00111_0000000000000000;    //addi r3 im(0) -> r7
ram[29] = 32'b001000_00100_00111_0000000000000000;    //addi r4 im(0) -> r7
ram[30] = 32'b001000_00101_00111_0000000000000000;    //addi r5 im(0) -> r7
```

最后这五行 addi 是后来为了方便看出排序的正确性加上的，在最后一张截图可以看到 r1 到 r5 的值是按顺序排好的。

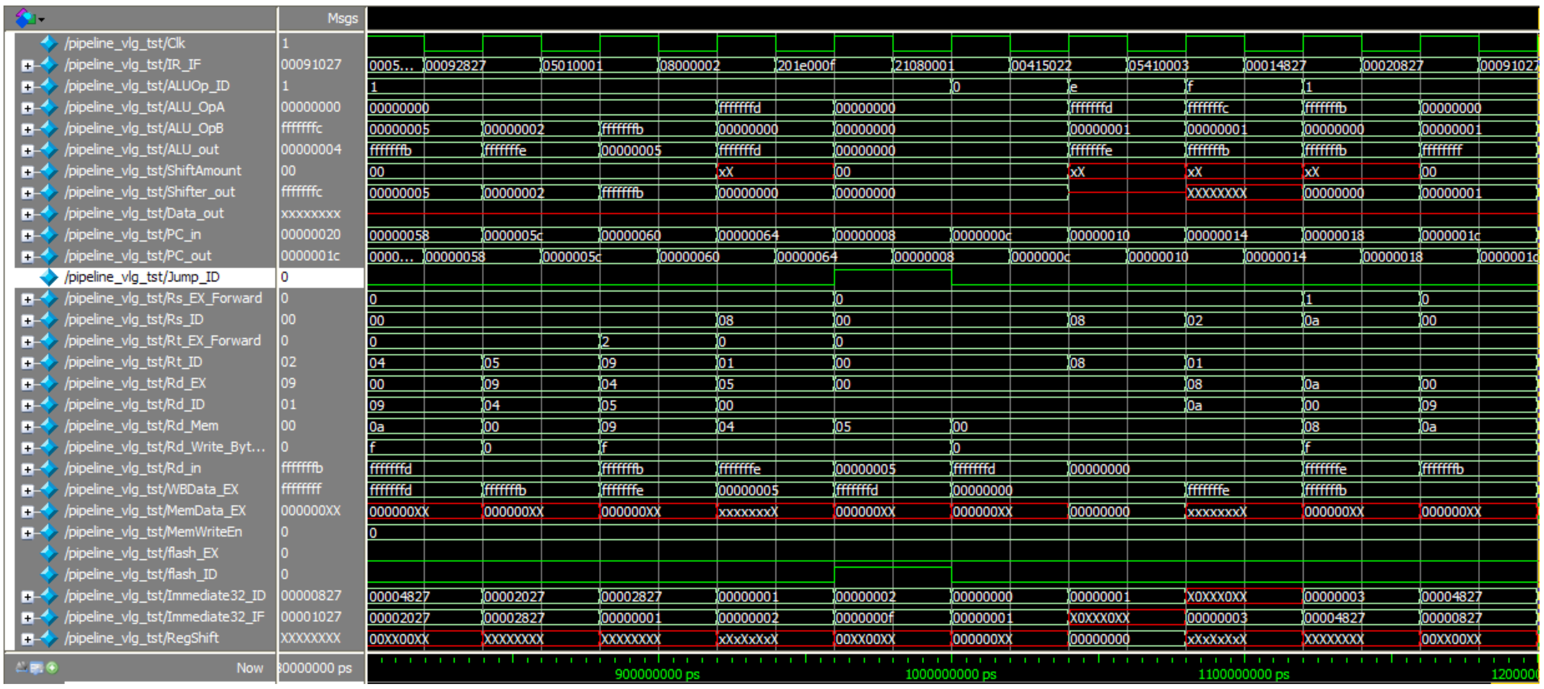
仿真截图：
每张图有十个周期，每张图的周期都不重复。



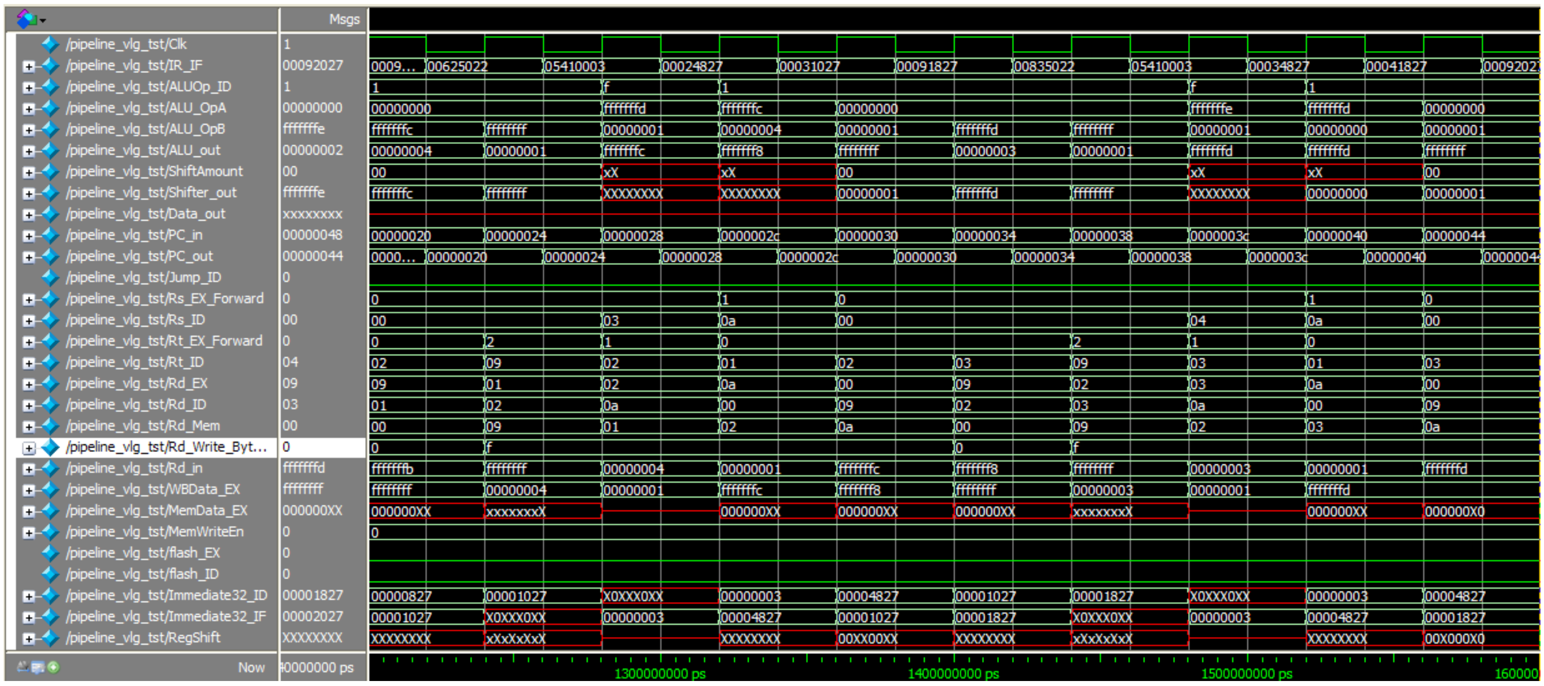
```
                                     Msgs
/pipeline_vlg_tst/Clk                1
/pipeline_vlg_tst/IR_IF              00031027
/pipeline_vlg_tst/ALUOp_ID           1
/pipeline_vlg_tst/ALU_OpA            00000000
/pipeline_vlg_tst/ALU_OpB            00000005
/pipeline_vlg_tst/ALU_out            fffffffb
/pipeline_vlg_tst/ShiftAmount        00
/pipeline_vlg_tst/Shifter_out        00000005
/pipeline_vlg_tst/Data_out           xxxxxxxx
/pipeline_vlg_tst/PC_in              00000030
/pipeline_vlg_tst/PC_out             0000002c
/pipeline_vlg_tst/Jump_ID            0
/pipeline_vlg_tst/Rs_EX_Forward      0
/pipeline_vlg_tst/Rs_ID              00
/pipeline_vlg_tst/Rt_EX_Forward      0
/pipeline_vlg_tst/Rt_ID              02
/pipeline_vlg_tst/Rd_EX              00
/pipeline_vlg_tst/Rd_ID              09
/pipeline_vlg_tst/Rd_Mem             0a
/pipeline_vlg_tst/Rd_Write_Byt...    f
/pipeline_vlg_tst/Rd_in              ffffffff
/pipeline_vlg_tst/WBData_EX          ffffffff
/pipeline_vlg_tst/MemData_EX         000000XX
/pipeline_vlg_tst/MemWriteEn         0
/pipeline_vlg_tst/flash_EX           0
/pipeline_vlg_tst/flash_ID           0
/pipeline_vlg_tst/Immediate32_ID     00004827
/pipeline_vlg_tst/Immediate32_IF     00001027
/pipeline_vlg_tst/RegShift           00XX00XX
Now                                  )0000000 ps
```



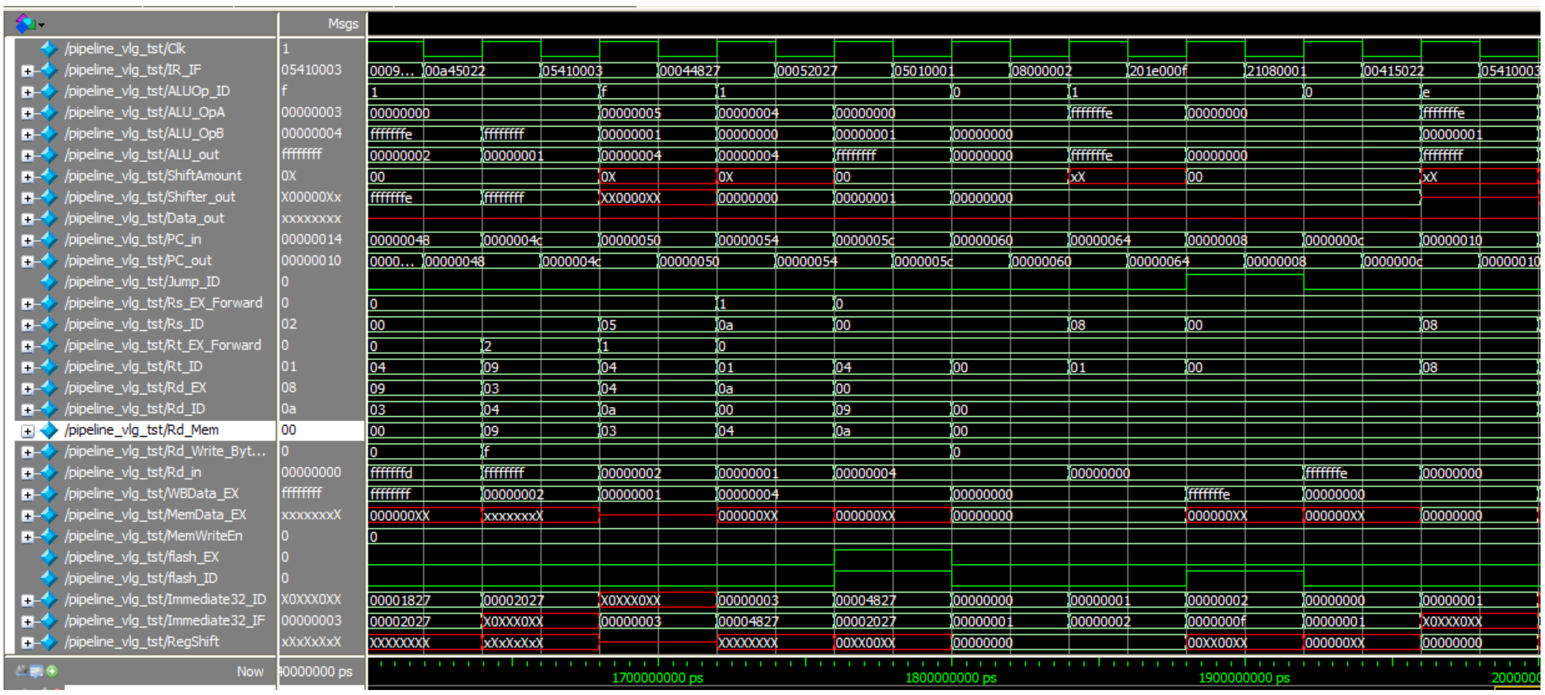
```
                                     Msgs
/pipeline_vlg_tst/Clk                1
/pipeline_vlg_tst/IR_IF              00052027
/pipeline_vlg_tst/ALUOp_ID           1
/pipeline_vlg_tst/ALU_OpA            00000000
/pipeline_vlg_tst/ALU_OpB            00000005
/pipeline_vlg_tst/ALU_out            fffffffb
/pipeline_vlg_tst/ShiftAmount        00
/pipeline_vlg_tst/Shifter_out        00000005
/pipeline_vlg_tst/Data_out           xxxxxxxx
/pipeline_vlg_tst/PC_in              00000058
/pipeline_vlg_tst/PC_out             00000054
/pipeline_vlg_tst/Jump_ID            0
/pipeline_vlg_tst/Rs_EX_Forward      0
/pipeline_vlg_tst/Rs_ID              00
/pipeline_vlg_tst/Rt_EX_Forward      0
/pipeline_vlg_tst/Rt_ID              04
/pipeline_vlg_tst/Rd_EX              00
/pipeline_vlg_tst/Rd_ID              09
/pipeline_vlg_tst/Rd_Mem             0a
/pipeline_vlg_tst/Rd_Write_Byt...    f
/pipeline_vlg_tst/Rd_in              fffffffd
/pipeline_vlg_tst/WBData_EX          fffffffd
/pipeline_vlg_tst/MemData_EX         000000XX
/pipeline_vlg_tst/MemWriteEn         0
/pipeline_vlg_tst/flash_EX           0
/pipeline_vlg_tst/flash_ID           0
/pipeline_vlg_tst/Immediate32_ID     00004827
/pipeline_vlg_tst/Immediate32_IF     00002027
/pipeline_vlg_tst/RegShift           00XX00XX
Now                                  )0000000 ps
```



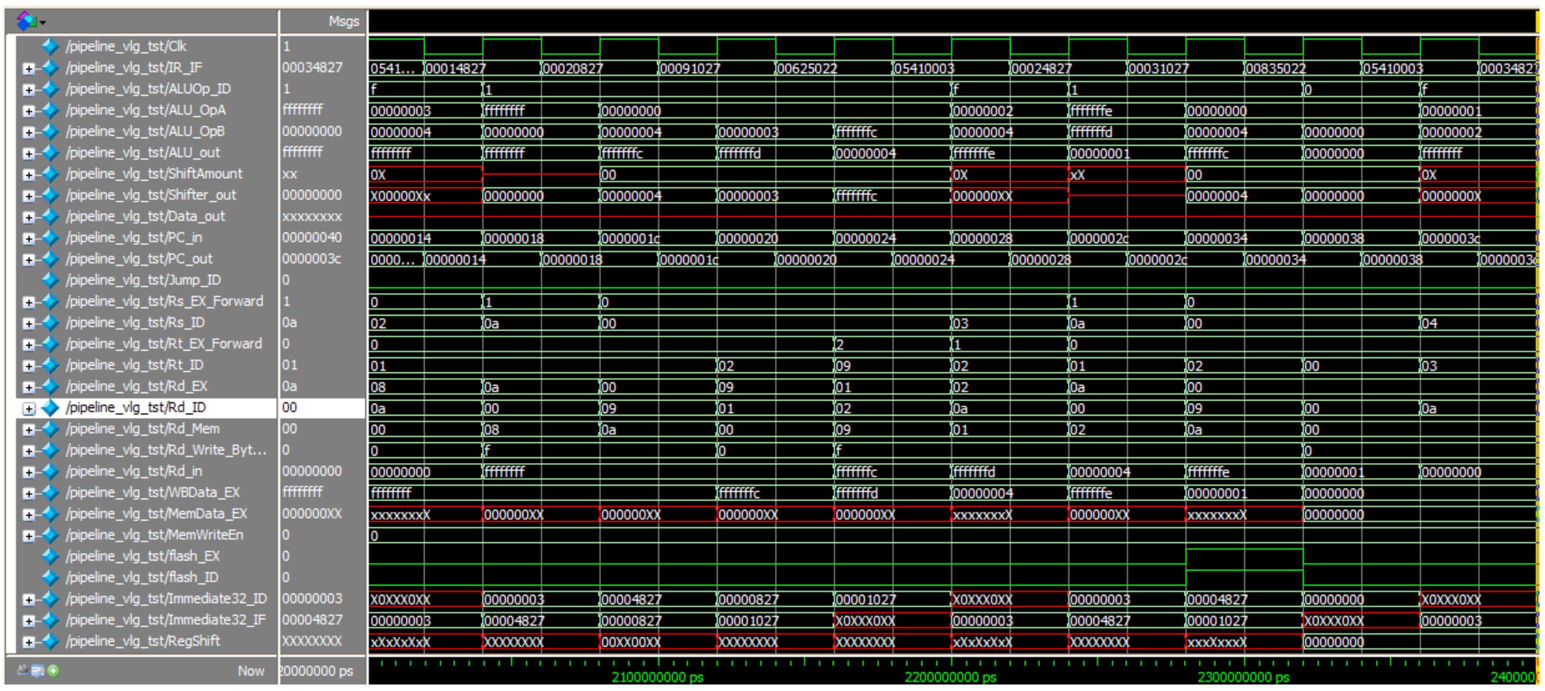
```
                                 Msgs
/pipeline_vlg_tst/Clk            1
/pipeline_vlg_tst/IR_IF          00091027
/pipeline_vlg_tst/ALUOp_ID       1
/pipeline_vlg_tst/ALU_OpA        00000000
/pipeline_vlg_tst/ALU_OpB        fffffffc
/pipeline_vlg_tst/ALU_out        00000004
/pipeline_vlg_tst/ShiftAmount    00
/pipeline_vlg_tst/Shifter_out    fffffffc
/pipeline_vlg_tst/Data_out       xxxxxxxx
/pipeline_vlg_tst/PC_in          00000020
/pipeline_vlg_tst/PC_out         0000001c
/pipeline_vlg_tst/Jump_ID        0
/pipeline_vlg_tst/Rs_EX_Forward  0
/pipeline_vlg_tst/Rs_ID          00
/pipeline_vlg_tst/Rt_EX_Forward  0
/pipeline_vlg_tst/Rt_ID          02
/pipeline_vlg_tst/Rd_EX          09
/pipeline_vlg_tst/Rd_ID          01
/pipeline_vlg_tst/Rd_Mem         00
/pipeline_vlg_tst/Rd_Write_Byt...  0
/pipeline_vlg_tst/Rd_in          fffffffb
/pipeline_vlg_tst/WBData_EX      ffffffff
/pipeline_vlg_tst/MemData_EX     000000XX
/pipeline_vlg_tst/MemWriteEn     0
/pipeline_vlg_tst/flash_EX       0
/pipeline_vlg_tst/flash_ID       0
/pipeline_vlg_tst/Immediate32_ID 00000827
/pipeline_vlg_tst/Immediate32_IF 00001027
/pipeline_vlg_tst/RegShift       XXXXXXXX
Now                              0000000 ps
```



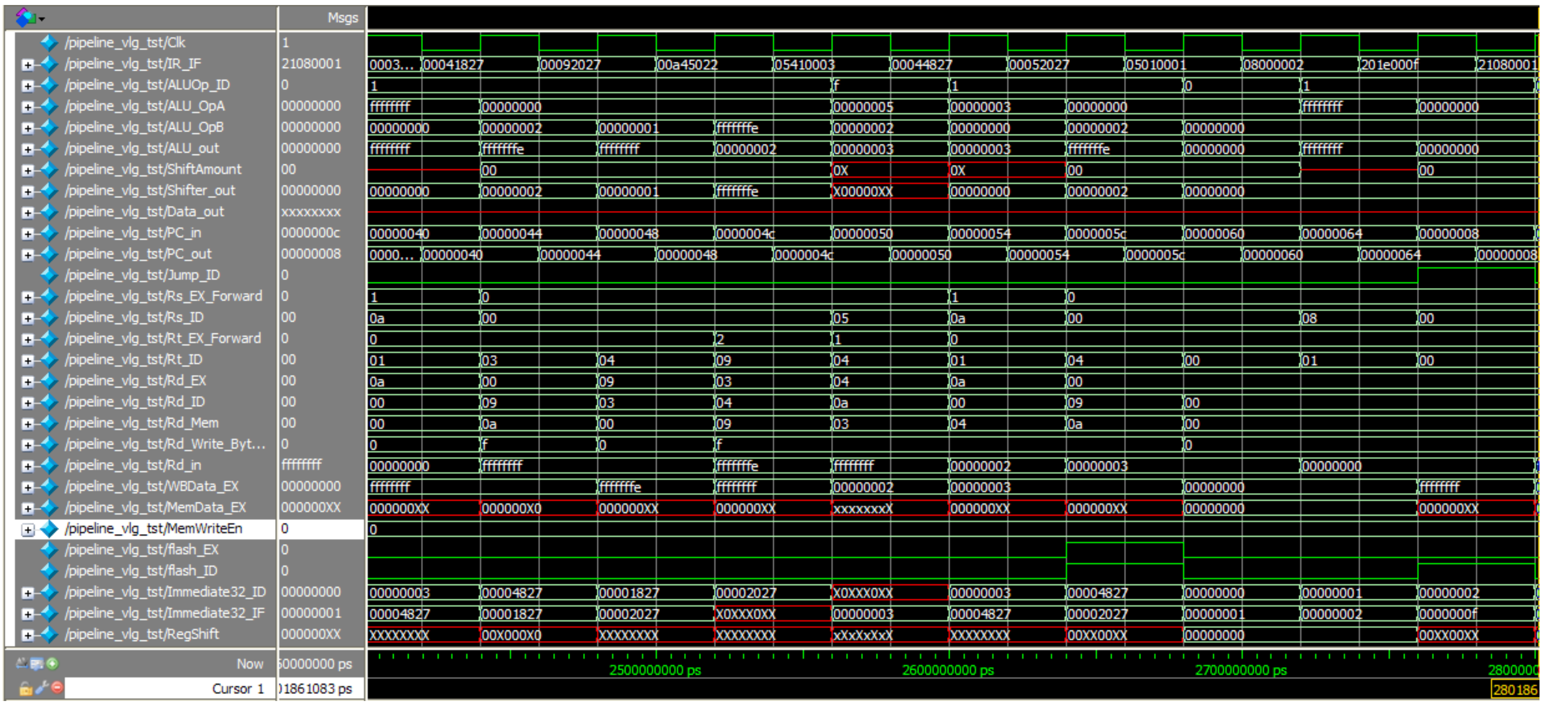
```
                                 Msgs
/pipeline_vlg_tst/Clk            1
/pipeline_vlg_tst/IR_IF          00092027
/pipeline_vlg_tst/ALUOp_ID       1
/pipeline_vlg_tst/ALU_OpA        00000000
/pipeline_vlg_tst/ALU_OpB        fffffffe
/pipeline_vlg_tst/ALU_out        00000002
/pipeline_vlg_tst/ShiftAmount    00
/pipeline_vlg_tst/Shifter_out    fffffffe
/pipeline_vlg_tst/Data_out       xxxxxxxx
/pipeline_vlg_tst/PC_in          00000048
/pipeline_vlg_tst/PC_out         00000044
/pipeline_vlg_tst/Jump_ID        0
/pipeline_vlg_tst/Rs_EX_Forward  0
/pipeline_vlg_tst/Rs_ID          00
/pipeline_vlg_tst/Rt_EX_Forward  0
/pipeline_vlg_tst/Rt_ID          04
/pipeline_vlg_tst/Rd_EX          09
/pipeline_vlg_tst/Rd_ID          03
/pipeline_vlg_tst/Rd_Mem         00
/pipeline_vlg_tst/Rd_Write_Byt...  0
/pipeline_vlg_tst/Rd_in          fffffffd
/pipeline_vlg_tst/WBData_EX      ffffffff
/pipeline_vlg_tst/MemData_EX     000000XX
/pipeline_vlg_tst/MemWriteEn     0
/pipeline_vlg_tst/flash_EX       0
/pipeline_vlg_tst/flash_ID       0
/pipeline_vlg_tst/Immediate32_ID 00001827
/pipeline_vlg_tst/Immediate32_IF 00002027
/pipeline_vlg_tst/RegShift       XXXXXXXX
Now                              0000000 ps
```



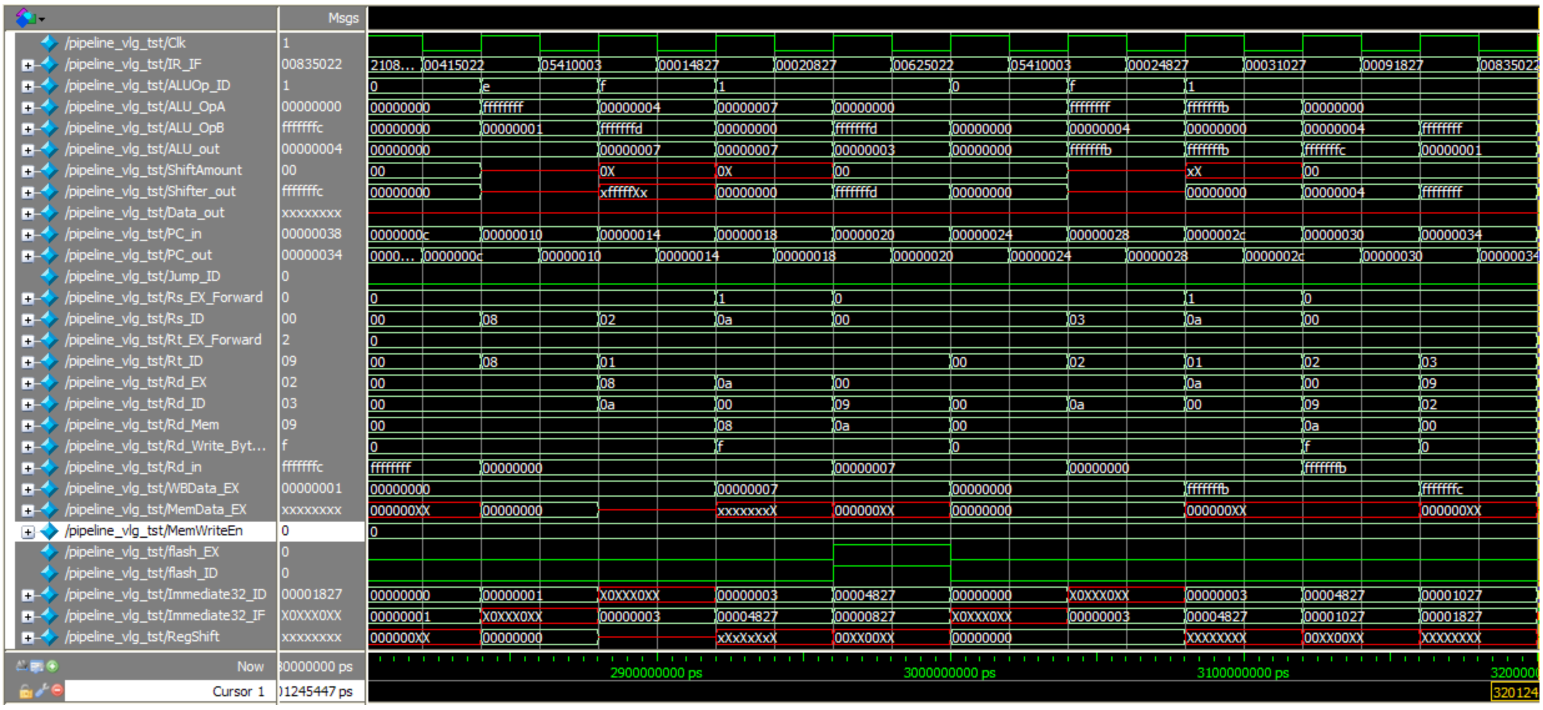
	Msgs
/pipeline_vlg_tst/Clk	1
/pipeline_vlg_tst/IR_IF	05410003
/pipeline_vlg_tst/ALUOp_ID	f
/pipeline_vlg_tst/ALU_OpA	00000003
/pipeline_vlg_tst/ALU_OpB	00000004
/pipeline_vlg_tst/ALU_out	ffffffff
/pipeline_vlg_tst/ShiftAmount	0X
/pipeline_vlg_tst/Shifter_out	X00000Xx
/pipeline_vlg_tst/Data_out	xxxxxxxx
/pipeline_vlg_tst/PC_in	00000014
/pipeline_vlg_tst/PC_out	00000010
/pipeline_vlg_tst/Jump_ID	0
/pipeline_vlg_tst/Rs_EX_Forward	0
/pipeline_vlg_tst/Rs_ID	02
/pipeline_vlg_tst/Rt_EX_Forward	0
/pipeline_vlg_tst/Rt_ID	01
/pipeline_vlg_tst/Rd_EX	08
/pipeline_vlg_tst/Rd_ID	0a
/pipeline_vlg_tst/Rd_Mem	00
/pipeline_vlg_tst/Rd_Write_Byt...	0
/pipeline_vlg_tst/Rd_in	00000000
/pipeline_vlg_tst/WBData_EX	ffffffff
/pipeline_vlg_tst/MemData_EX	xxxxxxxX
/pipeline_vlg_tst/MemWriteEn	0
/pipeline_vlg_tst/flash_EX	0
/pipeline_vlg_tst/flash_ID	0
/pipeline_vlg_tst/Immediate32_ID	X0XXX0XX
/pipeline_vlg_tst/Immediate32_IF	00000003
/pipeline_vlg_tst/RegShift	xXXxXXX
Now	0000000 ps



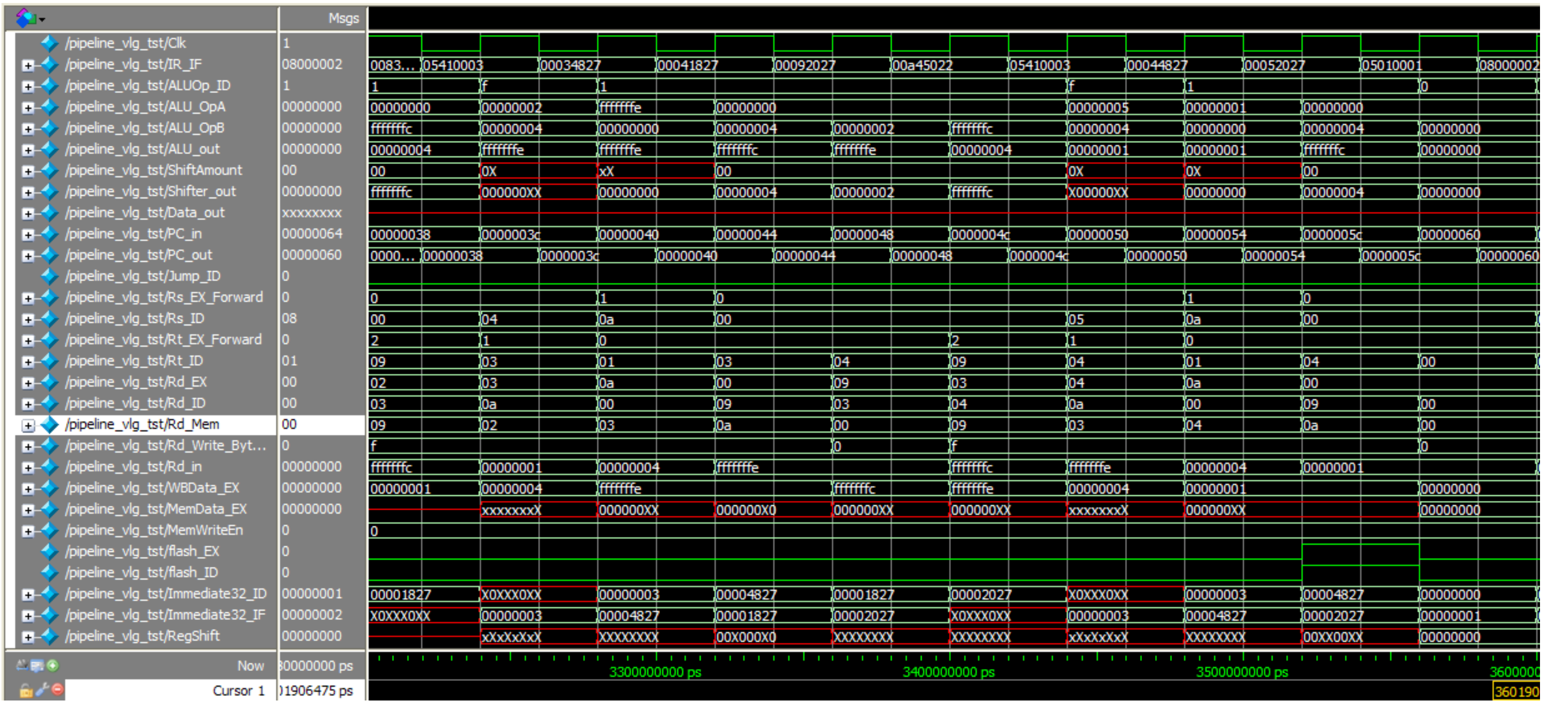
	Msgs
/pipeline_vlg_tst/Clk	1
/pipeline_vlg_tst/IR_IF	00034827
/pipeline_vlg_tst/ALUOp_ID	1
/pipeline_vlg_tst/ALU_OpA	ffffffff
/pipeline_vlg_tst/ALU_OpB	00000000
/pipeline_vlg_tst/ALU_out	ffffffff
/pipeline_vlg_tst/ShiftAmount	xx
/pipeline_vlg_tst/Shifter_out	00000000
/pipeline_vlg_tst/Data_out	xxxxxxxx
/pipeline_vlg_tst/PC_in	00000040
/pipeline_vlg_tst/PC_out	0000003c
/pipeline_vlg_tst/Jump_ID	0
/pipeline_vlg_tst/Rs_EX_Forward	1
/pipeline_vlg_tst/Rs_ID	0a
/pipeline_vlg_tst/Rt_EX_Forward	0
/pipeline_vlg_tst/Rt_ID	01
/pipeline_vlg_tst/Rd_EX	0a
/pipeline_vlg_tst/Rd_ID	00
/pipeline_vlg_tst/Rd_Mem	00
/pipeline_vlg_tst/Rd_Write_Byt...	0
/pipeline_vlg_tst/Rd_in	00000000
/pipeline_vlg_tst/WBData_EX	ffffffff
/pipeline_vlg_tst/MemData_EX	000000XX
/pipeline_vlg_tst/MemWriteEn	0
/pipeline_vlg_tst/flash_EX	0
/pipeline_vlg_tst/flash_ID	0
/pipeline_vlg_tst/Immediate32_ID	00000003
/pipeline_vlg_tst/Immediate32_IF	00004827
/pipeline_vlg_tst/RegShift	XXXXXXXX
Now	0000000 ps



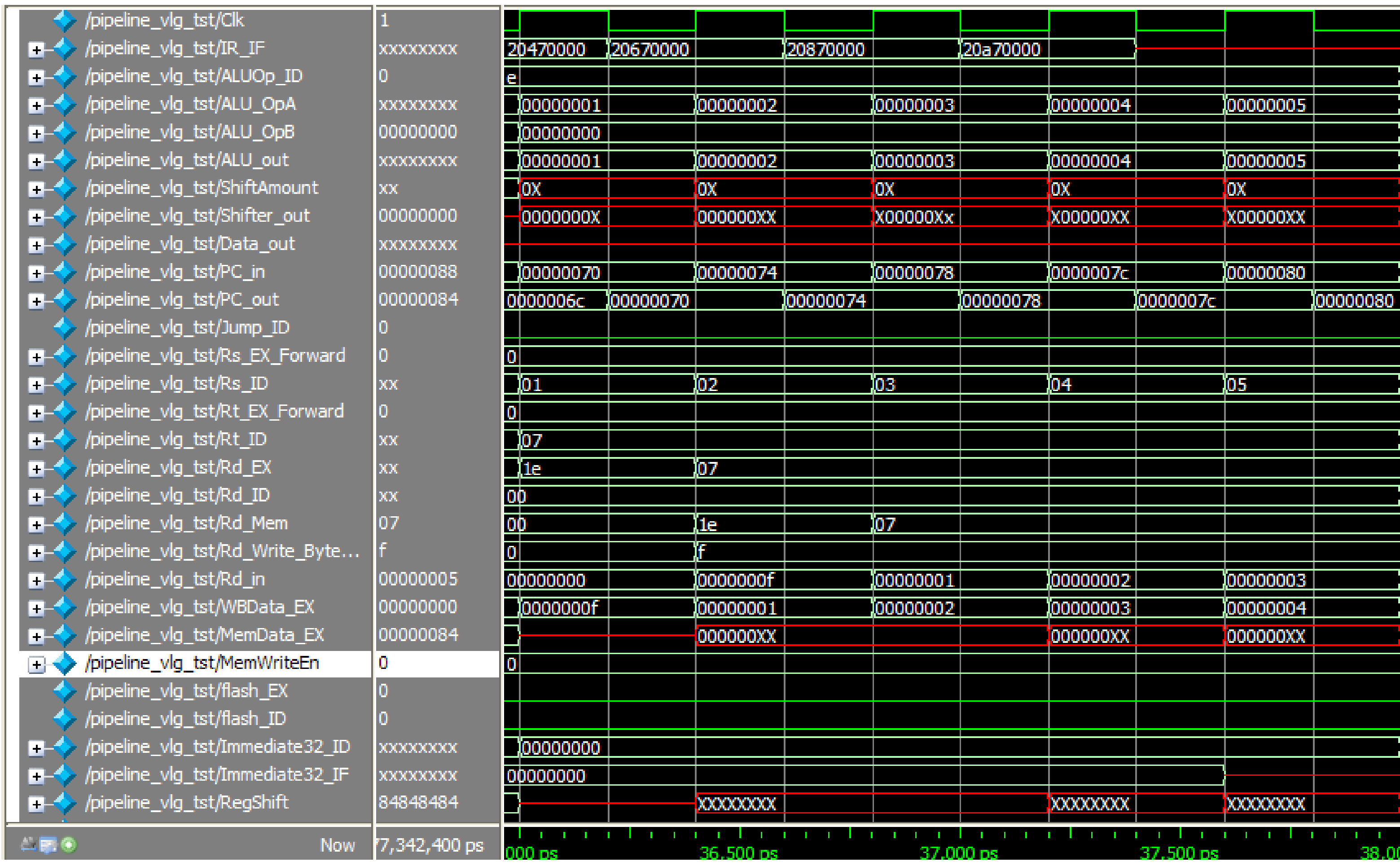
	Msgs
/pipeline_vlg_tst/Clk	1
/pipeline_vlg_tst/IR_IF	21080001
/pipeline_vlg_tst/ALUOp_ID	0
/pipeline_vlg_tst/ALU_OpA	00000000
/pipeline_vlg_tst/ALU_OpB	00000000
/pipeline_vlg_tst/ALU_out	00000000
/pipeline_vlg_tst/ShiftAmount	00
/pipeline_vlg_tst/Shifter_out	00000000
/pipeline_vlg_tst/Data_out	xxxxxxxx
/pipeline_vlg_tst/PC_in	0000000c
/pipeline_vlg_tst/PC_out	00000008
/pipeline_vlg_tst/Jump_ID	0
/pipeline_vlg_tst/Rs_EX_Forward	0
/pipeline_vlg_tst/Rs_ID	00
/pipeline_vlg_tst/Rt_EX_Forward	0
/pipeline_vlg_tst/Rt_ID	00
/pipeline_vlg_tst/Rd_EX	00
/pipeline_vlg_tst/Rd_ID	00
/pipeline_vlg_tst/Rd_Mem	00
/pipeline_vlg_tst/Rd_Write_Byt...	0
/pipeline_vlg_tst/Rd_in	ffffffff
/pipeline_vlg_tst/WBData_EX	00000000
/pipeline_vlg_tst/MemData_EX	000000XX
/pipeline_vlg_tst/MemWriteEn	0
/pipeline_vlg_tst/flash_EX	0
/pipeline_vlg_tst/flash_ID	0
/pipeline_vlg_tst/Immediate32_ID	00000000
/pipeline_vlg_tst/Immediate32_IF	00000001
/pipeline_vlg_tst/RegShift	000000XX
Now	0000000 ps
Cursor 1	1861083 ps



	Msgs
/pipeline_vlg_tst/Clk	1
/pipeline_vlg_tst/IR_IF	00835022
/pipeline_vlg_tst/ALUOp_ID	1
/pipeline_vlg_tst/ALU_OpA	00000000
/pipeline_vlg_tst/ALU_OpB	fffffffc
/pipeline_vlg_tst/ALU_out	00000004
/pipeline_vlg_tst/ShiftAmount	00
/pipeline_vlg_tst/Shifter_out	fffffffc
/pipeline_vlg_tst/Data_out	xxxxxxxx
/pipeline_vlg_tst/PC_in	00000038
/pipeline_vlg_tst/PC_out	00000034
/pipeline_vlg_tst/Jump_ID	0
/pipeline_vlg_tst/Rs_EX_Forward	0
/pipeline_vlg_tst/Rs_ID	00
/pipeline_vlg_tst/Rt_EX_Forward	2
/pipeline_vlg_tst/Rt_ID	09
/pipeline_vlg_tst/Rd_EX	02
/pipeline_vlg_tst/Rd_ID	03
/pipeline_vlg_tst/Rd_Mem	09
/pipeline_vlg_tst/Rd_Write_Byt...	f
/pipeline_vlg_tst/Rd_in	fffffffc
/pipeline_vlg_tst/WBData_EX	00000001
/pipeline_vlg_tst/MemData_EX	xxxxxxxx
/pipeline_vlg_tst/MemWriteEn	0
/pipeline_vlg_tst/flash_EX	0
/pipeline_vlg_tst/flash_ID	0
/pipeline_vlg_tst/Immediate32_ID	00001827
/pipeline_vlg_tst/Immediate32_IF	X0XXX0XX
/pipeline_vlg_tst/RegShift	xxxxxxxx
Now	0000000 ps
Cursor 1	1245447 ps



```
/pipeline_vlg_tst/Clk               1
/pipeline_vlg_tst/IR_IF             08000002
/pipeline_vlg_tst/ALUOp_ID          1
/pipeline_vlg_tst/ALU_OpA           00000000
/pipeline_vlg_tst/ALU_OpB           00000000
/pipeline_vlg_tst/ALU_out           00000000
/pipeline_vlg_tst/ShiftAmount       00
/pipeline_vlg_tst/Shifter_out       00000000
/pipeline_vlg_tst/Data_out          xxxxxxxx
/pipeline_vlg_tst/PC_in             00000064
/pipeline_vlg_tst/PC_out            00000060
/pipeline_vlg_tst/Jump_ID           0
/pipeline_vlg_tst/Rs_EX_Forward     0
/pipeline_vlg_tst/Rs_ID             08
/pipeline_vlg_tst/Rt_EX_Forward     0
/pipeline_vlg_tst/Rt_ID             01
/pipeline_vlg_tst/Rd_EX             00
/pipeline_vlg_tst/Rd_ID             00
/pipeline_vlg_tst/Rd_Mem            00
/pipeline_vlg_tst/Rd_Write_Byt...   0
/pipeline_vlg_tst/Rd_in             00000000
/pipeline_vlg_tst/WBData_EX         00000000
/pipeline_vlg_tst/MemData_EX        00000000
/pipeline_vlg_tst/MemWriteEn        0
/pipeline_vlg_tst/flash_EX          0
/pipeline_vlg_tst/flash_ID          0
/pipeline_vlg_tst/Immediate32_ID    00000001
/pipeline_vlg_tst/Immediate32_IF    00000002
/pipeline_vlg_tst/RegShift          00000000
Now       0000000 ps
Cursor 1  1906475 ps
```



```
/pipeline_vlg_tst/Clk               1
/pipeline_vlg_tst/IR_IF             xxxxxxxx
/pipeline_vlg_tst/ALUOp_ID          0
/pipeline_vlg_tst/ALU_OpA           xxxxxxxx
/pipeline_vlg_tst/ALU_OpB           00000000
/pipeline_vlg_tst/ALU_out           xxxxxxxx
/pipeline_vlg_tst/ShiftAmount       xx
/pipeline_vlg_tst/Shifter_out       00000000
/pipeline_vlg_tst/Data_out          xxxxxxxx
/pipeline_vlg_tst/PC_in             00000088
/pipeline_vlg_tst/PC_out            00000084
/pipeline_vlg_tst/Jump_ID           0
/pipeline_vlg_tst/Rs_EX_Forward     0
/pipeline_vlg_tst/Rs_ID             xx
/pipeline_vlg_tst/Rt_EX_Forward     0
/pipeline_vlg_tst/Rt_ID             xx
/pipeline_vlg_tst/Rd_EX             xx
/pipeline_vlg_tst/Rd_ID             xx
/pipeline_vlg_tst/Rd_Mem            07
/pipeline_vlg_tst/Rd_Write_Byte...  f
/pipeline_vlg_tst/Rd_in             00000005
/pipeline_vlg_tst/WBData_EX         00000000
/pipeline_vlg_tst/MemData_EX        00000084
/pipeline_vlg_tst/MemWriteEn        0
/pipeline_vlg_tst/flash_EX          0
/pipeline_vlg_tst/flash_ID          0
/pipeline_vlg_tst/Immediate32_ID    xxxxxxxx
/pipeline_vlg_tst/Immediate32_IF    xxxxxxxx
/pipeline_vlg_tst/RegShift          84848484
Now       7,342,400 ps
```

conclusion:

最后一张图看出，按顺序输出 r1~r5 的值为 1，2,3,4,5，所以，排序起效；