Project 2 Team Report

Group 9

November 12, 2020

Introduction to Computer Organization
VE370

Lu Dongyun	518370910171
Liang Jinghao	518370910014
Lu Mingxuan	518021911166
Zhang Zhixing	518370910067

University of Michigan-Shanghai Jiao Tong University Joint Institute(UM-SJTU JI)

Contents

1 Introduction **3**

2 Procedure **3**

2.1 Pipelined CPU Design 3

2.2 Strategy Logic to handle data hazards 3

2.3 Test Plan 3

3 Simulation **3**

3.1 Architecture without Data Hazard caused by beq instruction 3

3.2 Architecture with Data Hazard caused by beq instruction 3

4 Conclusion **3**

Appendix A: Complete Code **3**

Appendix B: RTL schematic of pipeline CPU **3**

1 Introduction

Compared to single cycle processors, processors with pipelined architecture have various advantages, including faster processing speed of instructions and smarter access to internal values. In this project, we will use Verilog to implement a pipelined processor which supports a subset of MIPS instructions set, including:

- R-type instructions: add, sub, and, or, slt
- item I-type instructions: lw, sw, addi, andi
- item J-type instructions: beq, bne, j

Besides, we will develop a method to resolve the data hazard caused by beq instructions. This requires extra control signal topology to enable this kind of data hazard. After the implementation of the processors, simulations of both original pipeline processor and the modified one which supports beq data hazard detection will be introduced to examine the functionality of the processors with regard of the MIPS instructions.

2 Procedure

2.1 Pipelined CPU Design

For a Pipelined CPU, we refer to the design shown in Figure 1. Our group's design of pipeline CPU is shown in Figure 2.

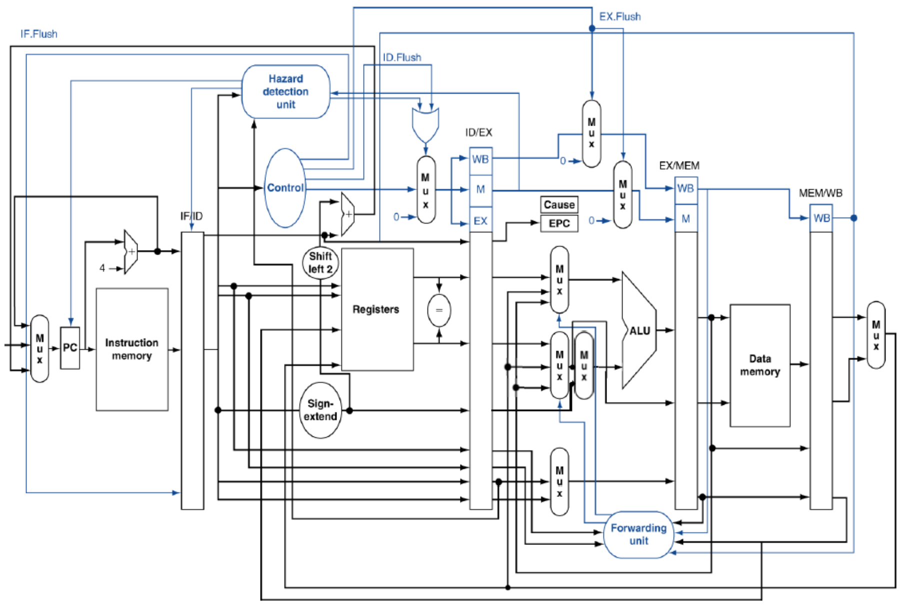


Figure 1. Design of Pipelined CPU

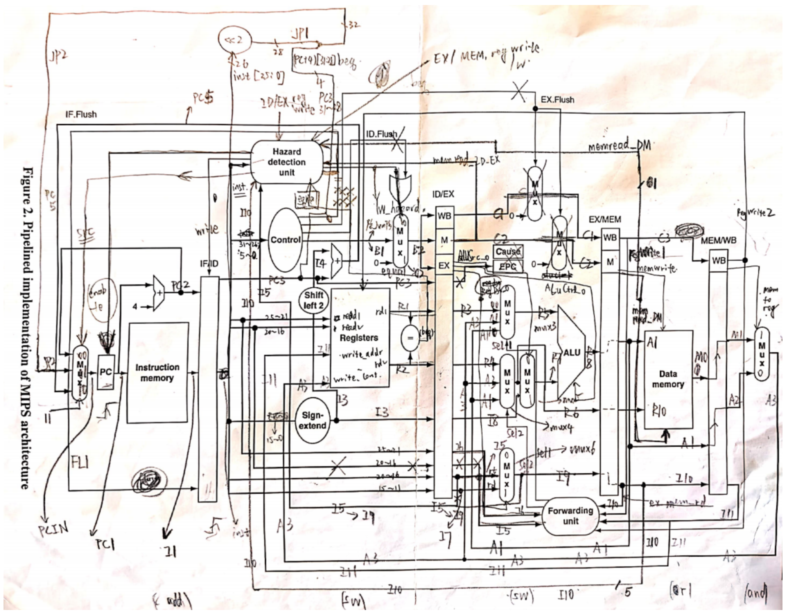


Figure 2. Our Group's Design of Pipelined CPU

To improve the performance of a single-cycle CPU, we divide the whole execution process into five stages: "IF", "ID", "EX", "MEM", "WB". To implement this new design, we insert four stage registers, "IF/ID", "ID/EX", "EX/MEM", and "MEM/WB" between any two stages. Such implementation will allow the CPU to process several instructions at the same time.

On one hand, it will improve the performance of the CPU by a large scale. On the other hand, there will be data hazard, control hazard and branch hazard due to the pipelined working pattern. To handle these problems, the CPU needs to be able to detect what kind of hazard exists and correspondingly forward or stall or flush.

Therefore, two blocks, called "**Hazard Detection**" and "**Forward**" respectively come in handy. Additionally, in our design, we **remove the module called "ALU control" and replace it with the signal "ALU Ctrl"** generate directly from the "Control" module. Additionally, the writing procedure only happens at negedge of clock, while reading can happen at any time.

Block Name	**Module Name**
ALU	alu.v
Adder	adder.v
Control	control.v
Instruction Memory	instmem.v
Mux	mux41.v, mux21.v
Registers	regfile.v
Data Memory	datamem.v
Sign-extend	extender.v
PC	reg.v
Shifter	shifter2.v, shifter.v
Hazard detection unit	Hazard_detection.v
Comparator	comparator.v
IF/ID Register	IF_ID_regf.v
ID/EX Register	ID_EX_regf.v
EX/MEM Register	EX_MEM_regf.v
MEM/WB Register	MEM_WB_regf.v
Forwarding unit	forwarding.v

Table 1. Block Name vs. Module Name

For simulation on FPGA board, we does the following change to the modules (Table 2).

Block Name	**Module Name**
Clock Divider	clock_divider_1.v, clock_divider_2.v
Hazard Detection Unit	Hazard_detection.v
Instruction Memory	instmem.v
PC	reg.v
Ring Counter	ring_counter.v
SSD Driver	ssd_driver_modify.v, ssd_driver.v

Table 2. Modified & Added Modules

With source codes shown in *Appendix A*, we can obtain RTL schematic of a pipeline CPU, which is in *Appendix B*.

2.2 Strategy Logic to handle data hazards

In hazard detection unit, we do the following:

First, we consider if such situation exists that beq or bne instruction is in ID stage, but Regwrite signal of the instruction in either EX stage or MEM stage equals to 1 and the Write register is the same as the read register of beq or bne. If it does exist, we stall beq in the ID stage until the instruction in EX stage or MEM stage finish writing back to the registers. We implement this function in the following code.

```
if (((ID_EX_regwrite==1'b1 && (ID_EX_rt==inst[25:21] ||ID_EX_rt==inst[20:16]))||
    (EX_MEM_regwrite==1'b1 && (EX_MEM_rd==inst[25:21] || EX_MEM_rd==inst[20:16]))) && (
    inst[31:26]==6'b000100 || inst[31:26]==6'b000101))
    begin
        IF_ID_enable=1'b0;
        lw_hazard=1'b1;
        src=2'b00;
        enable_pc=1'b0;
        IF_ID_flush=1'b0;
    end
```

Then we consider if such situation exists that there are arithmetic instructions, such as add, addi in ID stage and lw instruction in EX stage, and there is a data hazard. If it does exist, stall the instruction in ID stage for one clock cycle. We implement this function in the following code.

```
else if(ID_EX_memread==1'b1  && ( ID_EX_rt==inst[25:21] || ID_EX_rt==inst[20:16] ))  //lw
    begin
        IF_ID_enable=1'b0;
        lw_hazard=1'b1;
        src=2'b00;
        enable_pc=1'b0;
        IF_ID_flush=1'b0;
    end
```

Next we consider if there is a **taken** branch instruction. If so, we change the PC input to the address of the branch address and at next posedge, flush the wrong instruction right after the branch instruction by inserting a nop instruction after the branch instruction. We implement this function in the following code.

```
else if(inst[31:26]==6'b000100 && equal==1'b1) // beq
   begin
        IF_ID_flush=1'b1;
        src=2'b10;
        enable_pc=1'b1;
        IF_ID_enable=1'b1;
        lw_hazard=1'b0;
   end
   else if(inst[31:26]==6'b000101 && equal==1'b0)// bne
   begin
        IF_ID_flush=1'b1;
        src=2'b10;
        enable_pc=1'b1;
        IF_ID_enable=1'b1;
        lw_hazard=1'b0;
   end
```

Last but not least, we consider the situation including j instructions. If there is a j instruction in ID stage, we change the PC input to the address of the jump address and directly flush the instruction right after the j instruction at next posedge by inserting a nop instruction after the j instruction. After the nop instruction follows the instruction located at the jump address. We implement this function in the following code.

```
else if(inst[31:26]==6'b000010) //j
   begin
        IF_ID_flush=1'b1;
        src=2'b01;
        enable_pc=1'b1;
        IF_ID_enable=1'b1;
        lw_hazard=1'b0;
   end
```

What's left is the default situation(without detecting any hazard). The code for the default situation is as below.

```
else begin
        IF_ID_flush=1'b0;
        src=2'b00;
        enable_pc=1'b1;
        IF_ID_enable=1'b1;
        lw_hazard=1'b0;
   end
```

2.3 Test Plan

We divide the test into two parts. In the first part, we test instructions including **sw**, **lw**, **add**, **addi**, **sub**, **and**, **andi**, **or**, **slit**, **beq**, **bne** and **j**. We use the following instructions in "*InstructionMem_for_P2_Demo.txt*" in the first part.

```
00100000 00001000 00000000 00100000 //addi $t0, $zero, 0x20     00
00100000 00001001 00000000 00110111 //addi $t1, $zero, 0x37     04
00000001 00001001 10000000 00100100 //and $s0, $t0, $t1         08
00000001 00001001 10000000 00100101 //or $s0, $t0, $t1          0c
10101100 00010000 00000000 00000100 //sw $s0, 4($zero)          10
10101100 00001000 00000000 00001000 //sw $t0, 8($zero)          14
00000001 00001001 10001000 00100000 //add $s1, $t0, $t1         18
00000001 00001001 10010000 00100010 //sub $s2, $t0, $t1         1c
00100000 00001000 00000000 00100000 //addi $t0, $zero, 0x20     20
00100000 00001000 00000000 00100000 //addi $t0, $zero, 0x20     24
00100000 00001000 00000000 00100000 //addi $t0, $zero, 0x20     28
00010010 00110010 00000000 00010010 //beq $s1, $s2, error0      2c
10001100 00010001 00000000 00000100 //lw $s1, 4($zero)          30
00110010 00110010 00000000 01001000 //andi $s2, $s1, 0x48       34
00100000 00001000 00000000 00100000 //addi $t0, $zero, 0x20     38
00100000 00001000 00000000 00100000 //addi $t0, $zero, 0x20     3c
00100000 00001000 00000000 00100000 //addi $t0, $zero, 0x20     40
00010010 00110010 00000000 00001111 //beq $s1, $s2, error1      44
10001100 00010011 00000000 00001000 //lw $s3, 8($zero)          48
00100000 00001000 00000000 00100000 //addi $t0, $zero, 0x20     4c
00100000 00001000 00000000 00100000 //addi $t0, $zero, 0x20     50
00100000 00001000 00000000 00100000 //addi $t0, $zero, 0x20     54
00010010 00010011 00000000 00001101 //beq $s0, $s3, error2      58
00000010 01010001 10100000 00101010 //slt $s4, $s2, $s1         5c(Last)
00100000 00001000 00000000 00100000 //addi $t0, $zero, 0x20     60
00100000 00001000 00000000 00100000 //addi $t0, $zero, 0x20     64
00100000 00001000 00000000 00100000 //addi $t0, $zero, 0x20     68
00010010 10000000 00000000 00001111 //beq $s4, $0, EXIT         6c
00000010 00100000 10010000 00100000 //add $s2, $s1, $0          70
00001000 00000000 00000000 00010111 //j Last                    74
00100000 00001000 00000000 00000000 //addi $t0, $0, 0           78(error0)
00100000 00001001 00000000 00000000 //addi $t1, $0, 0           7c
00001000 00000000 00000000 00111111 //j EXIT                    80
00100000 00001000 00000000 00000001 //addi $t0, $0, 1           84(error1)
00100000 00001001 00000000 00000001 //addi $t1, $0, 1           88
00001000 00000000 00000000 00111111 //j EXIT                    8c
00100000 00001000 00000000 00000010 //addi $t0, $0, 2           90(error2)
00100000 00001001 00000000 00000010 //addi $t1, $0, 2           94
00001000 00000000 00000000 00111111 //j EXIT                    98
00100000 00001000 00000000 00000011 //addi $t0, $0, 3           9c(error3)
00100000 00001001 00000000 00000011 //addi $t1, $0, 3           a0
00001000 00000000 00000000 00111111 //j EXIT                    a4
```

We use the following instructions in "*InstructionMem_for_P2_Demo_bonus.txt*" i to check whether we solve the data hazards involving **beq** instruction in the second part.

```
00100000 00001000 00000000 00100000 //addi $t0, $zero, 0x20     00
00100000 00001001 00000000 00110111 //addi $t1, $zero, 0x37     04
00000001 00001001 10000000 00100100 //and $s0, $t0, $t1         08
00000001 00001001 10000000 00100101 //or $s0, $t0, $t1          0c
10101100 00010000 00000000 00000100 //sw $s0, 4($zero)          10
10101100 00001000 00000000 00001000 //sw $t0, 8($zero)          14
00000001 00001001 10001000 00100000 //add $s1, $t0, $t1         18
00000001 00001001 10010000 00100010 //sub $s2, $t0, $t1         1c
00010010 00110010 00000000 00001001 //beq $s1, $s2, error0      20
10001100 00010001 00000000 00000100 //lw $s1, 4($zero)          24
00110010 00110010 00000000 01001000 //andi $s2, $s1, 0x48       28
00010010 00110010 00000000 00001001 //beq $s1, $s2, error1      2c
10001100 00010011 00000000 00001000 //lw $s3, 8($zero)          30
00010010 00010011 00000000 00001010 //beq $s0, $s3, error2      34
00000010 01010001 10100000 00101010 //slt $s4, $s2, $s1         38(Last)
00010010 10000000 00000000 00001111 //beq $s4, $0, EXIT         3c
00000010 00100000 10010000 00100000 //add $s2, $s1, $0          40
00001000 00000000 00000000 00001110 //j Last                    44
00100000 00001000 00000000 00000000 //addi $t0, $0, 0           48(error0)
00100000 00001001 00000000 00000000 //addi $t1, $0, 0           4c
00001000 00000000 00000000 00011111 //j EXIT                    50
00100000 00001000 00000000 00000001 //addi $t0, $0, 1           54(error1)
00100000 00001001 00000000 00000001 //addi $t1, $0, 1           58
00001000 00000000 00000000 00011111 //j EXIT                    5c
00100000 00001000 00000000 00000010 //addi $t0, $0, 2           60(error2)
00100000 00001001 00000000 00000010 //addi $t1, $0, 2           64
00001000 00000000 00000000 00011111 //j EXIT                    68
00100000 00001000 00000000 00000011 //addi $t0, $0, 3           6c(error3)
00100000 00001001 00000000 00000011 //addi $t1, $0, 3           70
00001000 00000000 00000000 00011111 //j EXIT                    74
```

For the two part, we use the same set of verilog code except changing

```
$readmemb("InstructionMem_for_P2_Demo.txt",m1);
```

to

```
$readmemb("InstructionMem_for_P2_Demo_bonus.txt",m1);
```

The simulation code is as below:

```
module test_bench();
wire [31:0] pc, s0, s1, s2, s3, s4, s5, s6, s7, t0, t1, t2, t3, t4, t5, t6, t7, t8, t9;
reg clock;
reg [31:0] i;

pipeline_processor cpu1(pc, s0, s1,
         s2, s3, s4, s5, s6, s7, t0, t1, t2, t3, t4, t5, t6, t7, t8, t9, clock);

    initial begin
    #0 clock=0;
        $display("Time: %d, CLK= %b, PC= %h", 0, clock, pc);
        $display("[$s0]= %h, [$s1]= %h, [$s2]= %h,", s0, s1, s2);
        $display("[$s3]= %h, [$s4]= %h, [$s5]= %h,", s3, s4, s5);
        $display("[$s6]= %h, [$s7]= %h, [$t0]= %h,", s6, s7, t0);

        $display("[$t1]= %h, [$t2]= %h, [$t3]= %h,", t1, t2, t3);
        $display("[$t4]= %h, [$t5]= %h, [$t6]= %h,", t4, t5, t6);
        $display("[$t7]= %h, [$t8]= %h, [$t9]= %h,", t7, t8, t9);

        // THe first clock only lasts for half cycle
        // because we cannot initialize all the variables on posedge clock.
    #0.01
    for(i=0; i<=200; i=i+1)
    begin
        #10
        $display("Time: %d, CLK= %b, PC= %h", (i+1)*20 -10 , clock, pc);
        $display("[$s0]= %h, [$s1]= %h, [$s2]= %h,", s0, s1, s2);
        $display("[$s3]= %h, [$s4]= %h, [$s5]= %h,", s3, s4, s5);
        $display("[$s6]= %h, [$s7]= %h, [$t0]= %h,", s6, s7, t0);

        $display("[$t1]= %h, [$t2]= %h, [$t3]= %h,", t1, t2, t3);
        $display("[$t4]= %h, [$t5]= %h, [$t6]= %h,", t4, t5, t6);
        $display("[$t7]= %h, [$t8]= %h, [$t9]= %h,", t7, t8, t9);

        #10
        $display("Time: %d, CLK= %b, PC= %h", (i+1)*20, clock, pc);
        $display("[$s0]= %h, [$s1]= %h, [$s2]= %h,", s0, s1, s2);
        $display("[$s3]= %h, [$s4]= %h, [$s5]= %h,", s3, s4, s5);
        $display("[$s6]= %h, [$s7]= %h, [$t0]= %h,", s6, s7, t0);

        $display("[$t1]= %h, [$t2]= %h, [$t3]= %h,", t1, t2, t3);
        $display("[$t4]= %h, [$t5]= %h, [$t6]= %h,", t4, t5, t6);
        $display("[$t7]= %h, [$t8]= %h, [$t9]= %h,", t7, t8, t9);
    end
    end
    always #10 clock =~clock;
//    initial #690 $stop;//bonus
    initial #800 $stop;
endmodule
```

3 Simulation

3.1 Architecture without Data Hazard caused by beq instruction

Here is the simulation result when we read "*InstructionMem_for_P2_Demo.txt*".

For easier understanding, it is necessary to classify how to read through this result:

1. The *instruction on annotation(comment)* is which **changing the register (in WB stage)**.
2. The *PC* is of which in the "centering" the pipeline **(in IF stage)**.
3. Bubbles to solve data hazards will be marked by Dn: nop, and discussed after the result.

```
Time:           0, CLK= 0, PC= 00000000     //1st cycle
[$s0]= 00000000, [$s1]= 00000000, [$s2]= 00000000,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000000,
[$t1]= 00000000, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:          10, CLK= 1, PC= 00000004      //2nd cycle
[$s0]= 00000000, [$s1]= 00000000, [$s2]= 00000000,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000000,
[$t1]= 00000000, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:          20, CLK= 0, PC= 00000004
[$s0]= 00000000, [$s1]= 00000000, [$s2]= 00000000,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000000,
[$t1]= 00000000, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:          30, CLK= 1, PC= 00000008      //3rd cycle
[$s0]= 00000000, [$s1]= 00000000, [$s2]= 00000000,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000000,
[$t1]= 00000000, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:          40, CLK= 0, PC= 00000008
[$s0]= 00000000, [$s1]= 00000000, [$s2]= 00000000,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000000,
[$t1]= 00000000, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:          50, CLK= 1, PC= 0000000c      // 4th cycle
[$s0]= 00000000, [$s1]= 00000000, [$s2]= 00000000,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000000,
[$t1]= 00000000, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:          60, CLK= 0, PC= 0000000c
[$s0]= 00000000, [$s1]= 00000000, [$s2]= 00000000,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000000,
[$t1]= 00000000, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:         70, CLK= 1, PC= 00000010  //addi $t0, $zero, 0x20 reaches WB
[$s0]= 00000000, [$s1]= 00000000, [$s2]= 00000000,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000000,
[$t1]= 00000000, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:         80, CLK= 0, PC= 00000010
[$s0]= 00000000, [$s1]= 00000000, [$s2]= 00000000,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000000, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:         90, CLK= 1, PC= 00000014  //addi $t1, $zero, 0x37 reaches WB
[$s0]= 00000000, [$s1]= 00000000, [$s2]= 00000000,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000000, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        100, CLK= 0, PC= 00000014
[$s0]= 00000000, [$s1]= 00000000, [$s2]= 00000000,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        110, CLK= 1, PC= 00000018  //and $s0, $t0, $t1 reaches WB
[$s0]= 00000000, [$s1]= 00000000, [$s2]= 00000000,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        120, CLK= 0, PC= 00000018
[$s0]= 00000020, [$s1]= 00000000, [$s2]= 00000000,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        130, CLK= 1, PC= 0000001c  //or $s0, $t0, $t1 reaches WB
[$s0]= 00000020, [$s1]= 00000000, [$s2]= 00000000,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        140, CLK= 0, PC= 0000001c
[$s0]= 00000037, [$s1]= 00000000, [$s2]= 00000000,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        150, CLK= 1, PC= 00000020  //sw $s0, 4($zero) reaches WB
[$s0]= 00000037, [$s1]= 00000000, [$s2]= 00000000,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        160, CLK= 0, PC= 00000020
[$s0]= 00000037, [$s1]= 00000000, [$s2]= 00000000,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        170, CLK= 1, PC= 00000024  //sw $t0, 8($zero) reaches WB
[$s0]= 00000037, [$s1]= 00000000, [$s2]= 00000000,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        180, CLK= 0, PC= 00000024
[$s0]= 00000037, [$s1]= 00000000, [$s2]= 00000000,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        190, CLK= 1, PC= 00000028  //add $s1, $t0, $t1 reaches WB
```

```
[$s0]= 00000037, [$s1]= 00000000, [$s2]= 00000000,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:         200, CLK= 0, PC= 00000028
[$s0]= 00000037, [$s1]= 00000057, [$s2]= 00000000,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:         210, CLK= 1, PC= 0000002c  //sub $s2, $t0, $t1 reaches WB
[$s0]= 00000037, [$s1]= 00000057, [$s2]= 00000000,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:         220, CLK= 0, PC= 0000002c
[$s0]= 00000037, [$s1]= 00000057, [$s2]= ffffffe9,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:         230, CLK= 1, PC= 00000030  //addi $t0, $zero, 0x20 reaches WB
[$s0]= 00000037, [$s1]= 00000057, [$s2]= ffffffe9,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:         240, CLK= 0, PC= 00000030
[$s0]= 00000037, [$s1]= 00000057, [$s2]= ffffffe9,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:         250, CLK= 1, PC= 00000034  //addi $t0, $zero, 0x20 reaches WB
[$s0]= 00000037, [$s1]= 00000057, [$s2]= ffffffe9,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:         260, CLK= 0, PC= 00000034
[$s0]= 00000037, [$s1]= 00000057, [$s2]= ffffffe9,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:         270, CLK= 1, PC= 00000038  //addi $t0, $zero, 0x20 reaches WB
[$s0]= 00000037, [$s1]= 00000057, [$s2]= ffffffe9,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:         280, CLK= 0, PC= 00000038
[$s0]= 00000037, [$s1]= 00000057, [$s2]= ffffffe9,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:         290, CLK= 1, PC= 00000038  //beq $s1, $s2, error0 reaches WB
[$s0]= 00000037, [$s1]= 00000057, [$s2]= ffffffe9,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:         300, CLK= 0, PC= 00000038
[$s0]= 00000037, [$s1]= 00000057, [$s2]= ffffffe9,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:         310, CLK= 1, PC= 0000003c  //lw $s1, 4($zero) reaches WB
[$s0]= 00000037, [$s1]= 00000057, [$s2]= ffffffe9,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        320, CLK= 0, PC= 0000003c
[$s0]= 00000037, [$s1]= 00000037, [$s2]= ffffffe9,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        330, CLK= 1, PC= 00000040  //D1: nop reaches WB
[$s0]= 00000037, [$s1]= 00000037, [$s2]= ffffffe9,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        340, CLK= 0, PC= 00000040
[$s0]= 00000037, [$s1]= 00000037, [$s2]= ffffffe9,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        350, CLK= 1, PC= 00000044  //andi $s2, $s1, 0x48 reaches WB
[$s0]= 00000037, [$s1]= 00000037, [$s2]= ffffffe9,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        360, CLK= 0, PC= 00000044
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000000,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        370, CLK= 1, PC= 00000048  //addi $t0, $zero, 0x20 reaches WB
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000000,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        380, CLK= 0, PC= 00000048
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000000,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        390, CLK= 1, PC= 0000004c  //addi $t0, $zero, 0x20 reaches WB
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000000,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        400, CLK= 0, PC= 0000004c
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000000,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        410, CLK= 1, PC= 00000050  //addi $t0, $zero, 0x20 reaches WB
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000000,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        420, CLK= 0, PC= 00000050
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000000,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        430, CLK= 1, PC= 00000054  //beq $s1, $s2, error1 reaches WB
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000000,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        440, CLK= 0, PC= 00000054
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000000,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        450, CLK= 1, PC= 00000058  //lw $s3, 8($zero) reaches WB
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000000,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        460, CLK= 0, PC= 00000058
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000000,
[$s3]= 00000020, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        470, CLK= 1, PC= 0000005c  //addi $t0, $zero, 0x20 reaches WB
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000000,
[$s3]= 00000020, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        480, CLK= 0, PC= 0000005c
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000000,
[$s3]= 00000020, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        490, CLK= 1, PC= 00000060  //addi $t0, $zero, 0x20 reaches WB
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000000,
[$s3]= 00000020, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        500, CLK= 0, PC= 00000060
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000000,
[$s3]= 00000020, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        510, CLK= 1, PC= 00000064  //addi $t0, $zero, 0x20 reaches WB
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000000,
[$s3]= 00000020, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        520, CLK= 0, PC= 00000064
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000000,
[$s3]= 00000020, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        530, CLK= 1, PC= 00000068  //beq $s0, $s3, error2 reaches WB
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000000,
[$s3]= 00000020, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        540, CLK= 0, PC= 00000068
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000000,
[$s3]= 00000020, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        550, CLK= 1, PC= 0000006c  //slt $s4, $s2, $s1 (Last) reaches WB
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000000,
[$s3]= 00000020, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        560, CLK= 0, PC= 0000006c
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000000,
[$s3]= 00000020, [$s4]= 00000001, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        570, CLK= 1, PC= 00000070  //addi $t0, $zero, 0x20 reaches WB
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000000,
[$s3]= 00000020, [$s4]= 00000001, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        580, CLK= 0, PC= 00000070
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000000,
[$s3]= 00000020, [$s4]= 00000001, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        590, CLK= 1, PC= 00000074  //addi $t0, $zero, 0x20 reaches WB
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000000,
[$s3]= 00000020, [$s4]= 00000001, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        600, CLK= 0, PC= 00000074
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000000,
[$s3]= 00000020, [$s4]= 00000001, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        610, CLK= 1, PC= 00000078  //addi $t0, $zero, 0x20 reaches WB
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000000,
[$s3]= 00000020, [$s4]= 00000001, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        620, CLK= 0, PC= 00000078
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000000,
[$s3]= 00000020, [$s4]= 00000001, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        630, CLK= 1, PC= 0000005c  //beq $s4, $0, EXIT reaches WB
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000000,
[$s3]= 00000020, [$s4]= 00000001, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        640, CLK= 0, PC= 0000005c
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000000,
[$s3]= 00000020, [$s4]= 00000001, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        650, CLK= 1, PC= 00000060  //add $s2, $s1, $0 reaches WB
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000000,
[$s3]= 00000020, [$s4]= 00000001, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        660, CLK= 0, PC= 00000060
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000037,
[$s3]= 00000020, [$s4]= 00000001, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        670, CLK= 1, PC= 00000064  //j Last reaches WB
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000037,
[$s3]= 00000020, [$s4]= 00000001, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        680, CLK= 0, PC= 00000064
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000037,
[$s3]= 00000020, [$s4]= 00000001, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        690, CLK= 1, PC= 00000068  //D2: nop reaches WB
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000037,
[$s3]= 00000020, [$s4]= 00000001, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        700, CLK= 0, PC= 00000068
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000037,
[$s3]= 00000020, [$s4]= 00000001, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        710, CLK= 1, PC= 0000006c  //slt $s4, $s2, $s1 (Last) reaches WB
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000037,
[$s3]= 00000020, [$s4]= 00000001, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        720, CLK= 0, PC= 0000006c
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000037,
[$s3]= 00000020, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        730, CLK= 1, PC= 00000070  //addi $t0, $zero, 0x20 reaches WB
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000037,
[$s3]= 00000020, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        740, CLK= 0, PC= 00000070
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000037,
[$s3]= 00000020, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        750, CLK= 1, PC= 000000ac  //addi $t0, $zero, 0x20 reaches WB
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000037,
[$s3]= 00000020, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        760, CLK= 0, PC= 000000ac
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000037,
[$s3]= 00000020, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        770, CLK= 1, PC= 000000b0  //addi $t0, $zero, 0x20 reaches WB
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000037,
[$s3]= 00000020, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        780, CLK= 0, PC= 000000b0
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000037,
[$s3]= 00000020, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        790, CLK= 1, PC= 000000b4  //beq $s4, $0, EXIT reaches WB
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000037,
[$s3]= 00000020, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
//EXIT
```

Data Hazards (we define data hazard as situation in which bubbles is needed) in this part:

Data Hazard	**on**	**previous instruction**	**following instruction**	**# bubbles inserted**
D1	$1	lw $s1, 4($zero)	andi $s2, $s1, 0x48	1

Data Hazard	on	previous instruction	following instruction	# bubbles inserted
D2	-	j Last	slt $s4, $s2, $s1	1

The simulation result schemes are shown in Figure 3 and Figure 4.

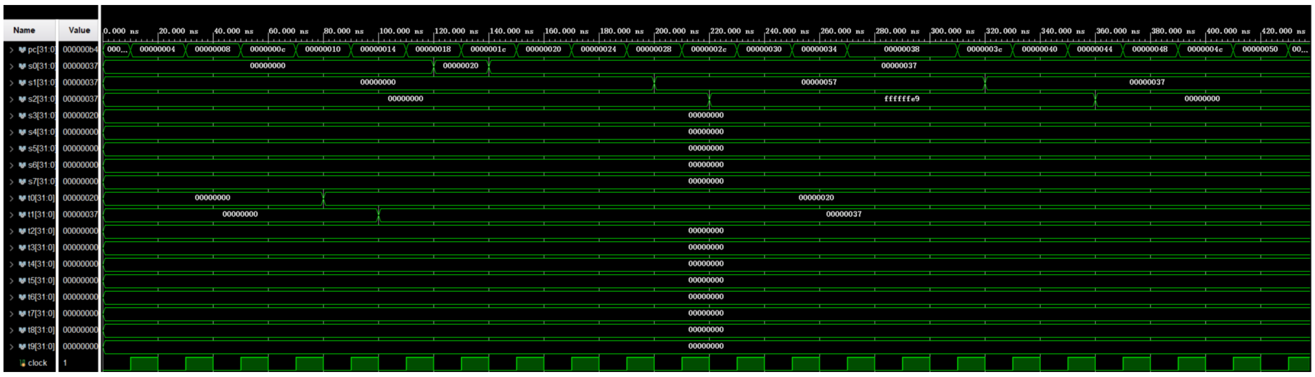


Figure 3. Simulation result scheme of architecture without beq data hazard 1

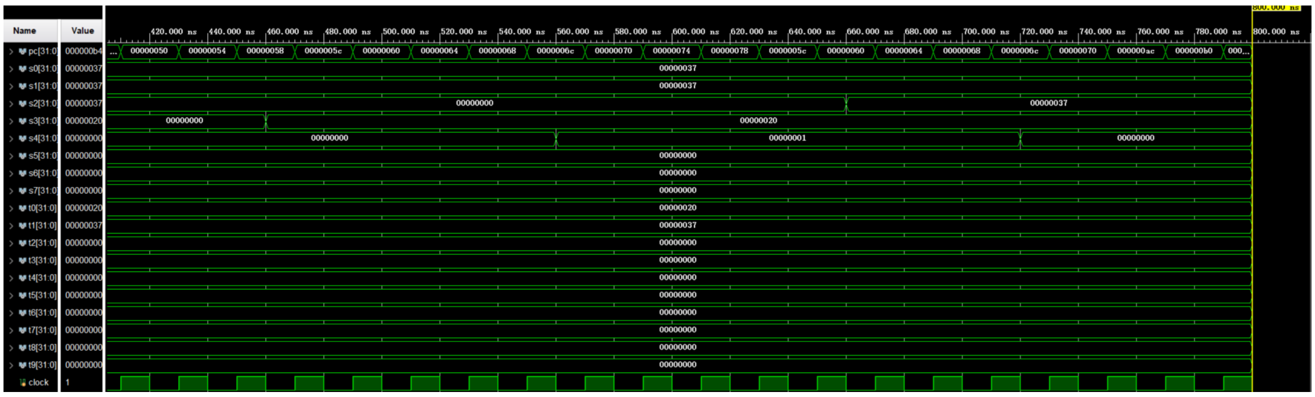


Figure 4. Simulation result scheme of architecture without beq data hazard 2

3.2 Architecture with Data Hazard caused by beq instruction

Here is the simulation result when we read "*InstructionMem_for_P2_Demo_bonus.txt*".

```
Time:          0, CLK= 0, PC= 00000000     //1st cycle
[$s0]= 00000000, [$s1]= 00000000, [$s2]= 00000000,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000000,
[$t1]= 00000000, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:         10, CLK= 1, PC= 00000004     //2st cycle
[$s0]= 00000000, [$s1]= 00000000, [$s2]= 00000000,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000000,
[$t1]= 00000000, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:         20, CLK= 0, PC= 00000004
[$s0]= 00000000, [$s1]= 00000000, [$s2]= 00000000,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000000,
[$t1]= 00000000, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:         30, CLK= 1, PC= 00000008     //3rd cycle
[$s0]= 00000000, [$s1]= 00000000, [$s2]= 00000000,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000000,
[$t1]= 00000000, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:         40, CLK= 0, PC= 00000008
[$s0]= 00000000, [$s1]= 00000000, [$s2]= 00000000,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000000,
[$t1]= 00000000, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:         50, CLK= 1, PC= 0000000c      //4th cycle
[$s0]= 00000000, [$s1]= 00000000, [$s2]= 00000000,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000000,
[$t1]= 00000000, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:         60, CLK= 0, PC= 0000000c
[$s0]= 00000000, [$s1]= 00000000, [$s2]= 00000000,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000000,
[$t1]= 00000000, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:         70, CLK= 1, PC= 00000010      //addi $t0, $zero, 0x20  reaches WB
[$s0]= 00000000, [$s1]= 00000000, [$s2]= 00000000,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000000,
[$t1]= 00000000, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:         80, CLK= 0, PC= 00000010
[$s0]= 00000000, [$s1]= 00000000, [$s2]= 00000000,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000000, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:         90, CLK= 1, PC= 00000014      //addi $t1, $zero, 0x37 reaches WB
[$s0]= 00000000, [$s1]= 00000000, [$s2]= 00000000,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000000, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        100, CLK= 0, PC= 00000014
[$s0]= 00000000, [$s1]= 00000000, [$s2]= 00000000,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        110, CLK= 1, PC= 00000018      //and $s0, $t0, $t1 reaches WB
[$s0]= 00000000, [$s1]= 00000000, [$s2]= 00000000,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        120, CLK= 0, PC= 00000018
[$s0]= 00000020, [$s1]= 00000000, [$s2]= 00000000,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        130, CLK= 1, PC= 0000001c      //or $s0, $t0, $t1 reaches WB
[$s0]= 00000020, [$s1]= 00000000, [$s2]= 00000000,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        140, CLK= 0, PC= 0000001c
[$s0]= 00000037, [$s1]= 00000000, [$s2]= 00000000,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        150, CLK= 1, PC= 00000020      //sw $s0, 4($zero) reaches WB
[$s0]= 00000037, [$s1]= 00000000, [$s2]= 00000000,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        160, CLK= 0, PC= 00000020
[$s0]= 00000037, [$s1]= 00000000, [$s2]= 00000000,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        170, CLK= 1, PC= 00000024      //sw $t0, 8($zero) reaches WB
[$s0]= 00000037, [$s1]= 00000000, [$s2]= 00000000,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        180, CLK= 0, PC= 00000024
[$s0]= 00000037, [$s1]= 00000000, [$s2]= 00000000,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        190, CLK= 1, PC= 00000024      //add $s1, $t0, $t1 reaches WB
[$s0]= 00000037, [$s1]= 00000000, [$s2]= 00000000,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        200, CLK= 0, PC= 00000024
[$s0]= 00000037, [$s1]= 00000057, [$s2]= 00000000,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        210, CLK= 1, PC= 00000024      //sub $s2, $t0, $t1 reaches WB
[$s0]= 00000037, [$s1]= 00000057, [$s2]= 00000000,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        220, CLK= 0, PC= 00000024
[$s0]= 00000037, [$s1]= 00000057, [$s2]= ffffffe9,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        230, CLK= 1, PC= 00000028      //D1: nop reaches WB
[$s0]= 00000037, [$s1]= 00000057, [$s2]= ffffffe9,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        240, CLK= 0, PC= 00000028
[$s0]= 00000037, [$s1]= 00000057, [$s2]= ffffffe9,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        250, CLK= 1, PC= 0000002c      //D1: nop reaches WB
[$s0]= 00000037, [$s1]= 00000057, [$s2]= ffffffe9,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        260, CLK= 0, PC= 0000002c
[$s0]= 00000037, [$s1]= 00000057, [$s2]= ffffffe9,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        270, CLK= 1, PC= 0000002c      //beq $s1, $s2, error0 reaches WB
[$s0]= 00000037, [$s1]= 00000057, [$s2]= ffffffe9,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        280, CLK= 0, PC= 0000002c
[$s0]= 00000037, [$s1]= 00000057, [$s2]= ffffffe9,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        290, CLK= 1, PC= 00000030      //lw $s1, 4($zero) reaches WB
[$s0]= 00000037, [$s1]= 00000057, [$s2]= ffffffe9,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        300, CLK= 0, PC= 00000030
[$s0]= 00000037, [$s1]= 00000037, [$s2]= ffffffe9,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        310, CLK= 1, PC= 00000030      //D2: nop reaches WB
[$s0]= 00000037, [$s1]= 00000037, [$s2]= ffffffe9,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        320, CLK= 0, PC= 00000030
[$s0]= 00000037, [$s1]= 00000037, [$s2]= ffffffe9,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        330, CLK= 1, PC= 00000030      //andi $s2, $s1, 0x48 reaches WB
[$s0]= 00000037, [$s1]= 00000037, [$s2]= ffffffe9,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        340, CLK= 0, PC= 00000030
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000000,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        350, CLK= 1, PC= 00000034      //D3: nop reaches WB
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000000,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        360, CLK= 0, PC= 00000034
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000000,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        370, CLK= 1, PC= 00000038      //D3: nop reaches WB
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000000,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        380, CLK= 0, PC= 00000038
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000000,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        390, CLK= 1, PC= 00000038      //beq $s1, $s2, error1 reaches WB
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000000,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        400, CLK= 0, PC= 00000038
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000000,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        410, CLK= 1, PC= 00000038      //lw $s3, 8($zero) reaches WB
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000000,
[$s3]= 00000000, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        420, CLK= 0, PC= 00000038
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000000,
[$s3]= 00000020, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        430, CLK= 1, PC= 0000003c      //D4: nop reaches WB
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000000,
[$s3]= 00000020, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        440, CLK= 0, PC= 0000003c
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000000,
[$s3]= 00000020, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        450, CLK= 1, PC= 00000040      //D4: nop reaches WB
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000000,
[$s3]= 00000020, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        460, CLK= 0, PC= 00000040
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000000,
[$s3]= 00000020, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        470, CLK= 1, PC= 00000040      //beq $s0, $s3, error2 reaches WB
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000000,
[$s3]= 00000020, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        480, CLK= 0, PC= 00000040
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000000,
[$s3]= 00000020, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        490, CLK= 1, PC= 00000040      //slt $s4, $s2, $s1 reaches WB
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000000,
[$s3]= 00000020, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        500, CLK= 0, PC= 00000040
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000000,
[$s3]= 00000020, [$s4]= 00000001, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        510, CLK= 1, PC= 00000044      //D5: nop reaches WB
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000000,
[$s3]= 00000020, [$s4]= 00000001, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        520, CLK= 0, PC= 00000044
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000000,
[$s3]= 00000020, [$s4]= 00000001, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        530, CLK= 1, PC= 00000048      //D5: nop reaches WB
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000000,
[$s3]= 00000020, [$s4]= 00000001, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        540, CLK= 0, PC= 00000048
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000000,
[$s3]= 00000020, [$s4]= 00000001, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        550, CLK= 1, PC= 00000038      //beq $s4, $0, EXIT reaches WB
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000000,
[$s3]= 00000020, [$s4]= 00000001, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        560, CLK= 0, PC= 00000038
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000000,
[$s3]= 00000020, [$s4]= 00000001, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        570, CLK= 1, PC= 0000003c      //add $s2, $s1, $0 reaches WB
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000000,
[$s3]= 00000020, [$s4]= 00000001, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        580, CLK= 0, PC= 0000003c
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000037,
[$s3]= 00000020, [$s4]= 00000001, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        590, CLK= 1, PC= 00000040      //j Last reaches WB
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000037,
[$s3]= 00000020, [$s4]= 00000001, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        600, CLK= 0, PC= 00000040
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000037,
[$s3]= 00000020, [$s4]= 00000001, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        610, CLK= 1, PC= 00000040      //D6: nop reaches WB
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000037,
[$s3]= 00000020, [$s4]= 00000001, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        620, CLK= 0, PC= 00000040
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000037,
[$s3]= 00000020, [$s4]= 00000001, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        630, CLK= 1, PC= 00000040      //slt $s4, $s2, $s1 reaches WB
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000037,
[$s3]= 00000020, [$s4]= 00000001, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        640, CLK= 0, PC= 00000040
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000037,
[$s3]= 00000020, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        650, CLK= 1, PC= 0000007c      //D7: nop reaches WB
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000037,
[$s3]= 00000020, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        660, CLK= 0, PC= 0000007c
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000037,
[$s3]= 00000020, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        670, CLK= 1, PC= 00000080      //D7: nop reaches WB
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000037,
[$s3]= 00000020, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        680, CLK= 0, PC= 00000080
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000037,
[$s3]= 00000020, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        690, CLK= 1, PC= 00000084      //beq $s4, $0, EXIT reaches WB
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000037,
[$s3]= 00000020, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        700, CLK= 0, PC= 00000084
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000037,
[$s3]= 00000020, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        710, CLK= 1, PC= 00000088      //EXIT: Useless
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000037,
[$s3]= 00000020, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        720, CLK= 0, PC= 00000088
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000037,
[$s3]= 00000020, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        730, CLK= 1, PC= 0000008c      //useless
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000037,
[$s3]= 00000020, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        740, CLK= 0, PC= 0000008c
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000037,
[$s3]= 00000020, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        750, CLK= 1, PC= 00000090      //useless
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000037,
[$s3]= 00000020, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        760, CLK= 0, PC= 00000090
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000037,
[$s3]= 00000020, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        770, CLK= 1, PC= 00000094      //useless
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000037,
[$s3]= 00000020, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        780, CLK= 0, PC= 00000094
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000037,
[$s3]= 00000020, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
Time:        790, CLK= 1, PC= 00000098      //useless
```

```
[$s0]= 00000037, [$s1]= 00000037, [$s2]= 00000037,
[$s3]= 00000020, [$s4]= 00000000, [$s5]= 00000000,
[$s6]= 00000000, [$s7]= 00000000, [$t0]= 00000020,
[$t1]= 00000037, [$t2]= 00000000, [$t3]= 00000000,
[$t4]= 00000000, [$t5]= 00000000, [$t6]= 00000000,
[$t7]= 00000000, [$t8]= 00000000, [$t9]= 00000000,
```

Data Hazards (we define the data hazard as situation in which bubbles is needed) in this part:

Data Hazard	on	previous instruction	following instruction	# bubbles inserted
D1	**$s2**	**sub $s2, $t0, $t1**	**beq $s1, $s2, error0**	**2**
D2	$s1	lw $s1, 4($zero)	andi $s2, $s1, 0x48	1
D3	**$s2**	**andi $s2, $s1, 0x48**	**beq $s1, $s2, error1**	**2**
D4	**$s3**	**lw $s3, 8($zero)**	**beq $s0, $s3, error2**	**2**
D5	**$s4**	**slt $s4, $s2, $s1**	**beq $s4, $0, EXIT**	**2**
D6	-	j Last	slt $s4, $s2, $s1	1
D7	**$s4**	**slt $s4, $s2, $s1**	**beq $s4, $0, EXIT**	**2**

The simulation result schemes are shown in Figure 5 and Figure 6.

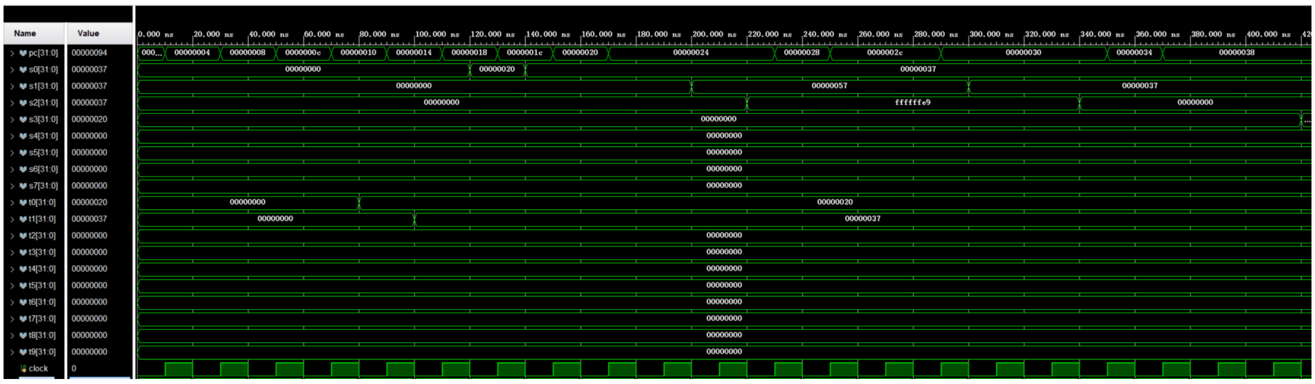


Figure 5. Simulation result scheme of architecture with beq data hazard 1

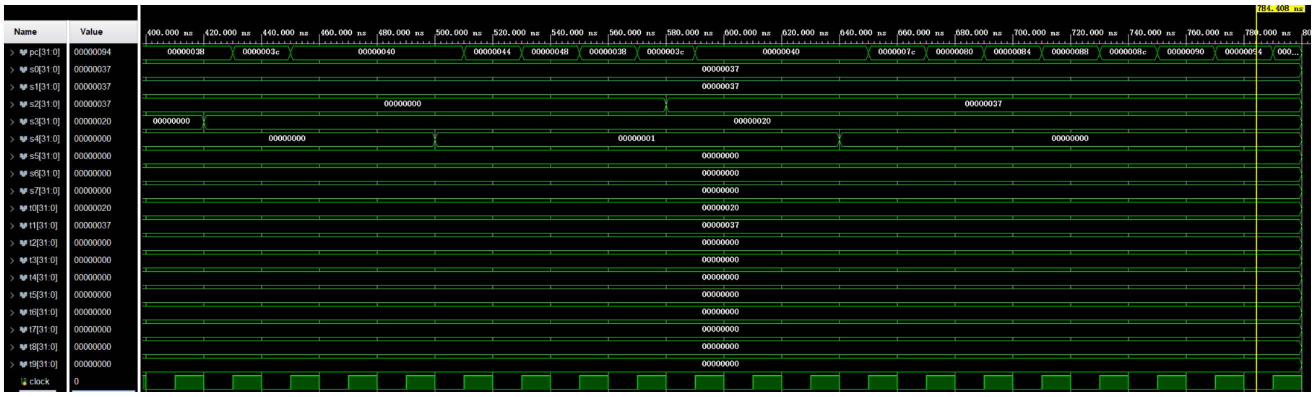


Figure 6. Simulation result scheme of architecture with beq data hazard 2

4 Conclusion

From the simulation results in section 3.1 we could see that the hazard detection, forwarding, control as well as ALU Ctrl units are working properly, as expected. When data hazards occurs, bubbles can be inserted for correct cycles of time. The control unit is able to generate accurate control signals to parts of the pipelined processor, according to the instructions provided.

As an additional design, the processor is also able to support the detection of data hazards caused by beq instructions. For this kind of data hazard, the processor will generate nops between instructions to resolve it. Section 3.2 shows that these data hazards were resolved correctly.

Appendix A: Complete Code

pipeline.v

```
module pipeline_processor(pc, s0, s1,
         s2, s3, s4, s5, s6, s7, t0, t1, t2, t3, t4, t5, t6, t7, t8, t9, clk);
input clk;
output [31:0] pc, s0, s1, s2, s3, s4, s5, s6, s7, t0, t1, t2, t3, t4, t5, t6, t7, t8, t9;


wire FL1, enable, write, equal, lw_hazard,
memread_ID_EX, memread_DM, regwrite1, memwrite, memtoreg, regwrite2;
wire [1:0] src, C1, C3, sel1, sel2;
wire [2:0] C2;
wire [4:0] I11, I5, I9, I10, I6, I7;
wire [10:0] B2;
wire [27:0] JP1;
wire [31:0] JP2, PC5, PC2, PC1, PCIN, I1, inst,
PC3, I3, I4, R1, R2, R3, A3, A1, R4, I8,
 R6, R7, R8, R5, R10, M0, M1, A2;

mux41 #32 mux1(PCIN, PC2, JP2, PC5, 0, src);
register regpc(PC1,PCIN,clk,enable);
adder adder1(PC2, PC1, 4);
inst_mem instruction_memory(I1, PC1);

IF_ID_regf IF_ID_regf1(PC3, inst, PC2, I1, write, FL1, clk);

shifter1 shifterb(JP1, inst[25:0]);


assign JP2={PC3[31:28], JP1};


wire RegDst, Jump, Branch, MemRead, MemtoReg_from_control, MemWrite, ALUSrc, RegWrite;
wire [3:0] AluCtrl;
controller controller1(RegDst, Jump, Branch, MemRead,
MemtoReg_from_control, AluCtrl, MemWrite, ALUSrc, RegWrite, inst[31:26], inst[5:0]);
mux21 #11 mux2(B2,
{MemtoReg_from_control, RegWrite, Branch, MemWrite, MemRead, RegDst, AluCtrl, ALUSrc},0,lw_hazard);
extender extendera(I3, inst[15:0]);
shifter2 shiftera(I4, I3);
adder adder2(PC5, I4, PC3);
register_file register_file1(s0, s1,
         s2, s3, s4, s5, s6, s7, t0, t1, t2, t3, t4, t5, t6, t7, t8, t9,
         R1, R2, regwrite2, inst[25:21], inst[20:16],
         I11, A3, clk);
comparator comparator1(equal, R1, R2);

wire MemtoReg_o, RegWrite_o, Branch_o, MemWrite_o, RegDst_o, ALUSrc_o;


hazard_detection hazard_detection_unit(src, enable, write, FL1, lw_hazard,
MemRead_ID_EX, MemRead_DM, inst , I9, I10, RegWrite_o, regwrite1, equal, clk);

assign pc=PC1;




wire [3:0] AluCtrl_o;

ID_EX_regf ID_EX_regf1(MemtoReg_o, RegWrite_o, Branch_o, MemWrite_o,
MemRead_ID_EX, RegDst_o, AluCtrl_o, ALUSrc_o, R3, R4,
 I8, I6, I5, I7,
 B2[10], B2[9], B2[8], B2[7], B2[6], B2[5],
 B2[4:1], B2[0], R1, R2, I3, inst[25:21], inst[20:16], inst[15:11], clk);




mux41 #32 mux3(R5,R3,A3,A1,0,sel1);
mux41 #32 mux4(R6,R4,A3,A1,0,sel2);
mux21 #32 mux5(R7, R6, I8, ALUSrc_o);
alusimple alu1(R8, zero, ovf, R5, R7, AluCtrl_o);
mux21 #5 mux6(I9, I5, I7, RegDst_o);
forwarding forwarding_u(sel1, sel2, I6, I5, I10, regwrite1, I11, regwrite2);


wire MemtoReg_EXMEM, RegWrite_EXMEM, Branch_EXMEM;
```

```
EX_MEM_regf EX_MEM_regf1(MemtoReg_EXMEM, regwrite1, Branch_EXMEM, memwrite,
MemRead_DM, A1, R10, I10,
MemtoReg_o, RegWrite_o, Branch_o, MemWrite_o,
MemRead_ID_EX, R8, R6, I9, clk);



data_mem date_memory_1(M0, memwrite, MemRead_DM, A1, R10, clk);


MEM_WB_regf MEM_WB_regf1(memtoreg, regwrite2, M1, A2, I11,
MemtoReg_EXMEM,  regwrite1, M0, A1, I10, clk);

mux21 #32 mux_WB(A3,A2,M1,memtoreg);


endmodule
```

adder.v

```
module adder(sum, i0, i1);
input [31:0] i0, i1;
output [31:0] sum;

assign sum = i0+i1;


endmodule
```

alu.v

```
module alusimple(result, zero, ovf, a, b, alu_ctrl);

input [3:0] alu_ctrl;
input [31:0] a, b;
output [31:0] result;
output zero, ovf;

reg [31:0] result;

always @(alu_ctrl or a or b) begin
if(alu_ctrl==4'b0000) result=a & b;
else if(alu_ctrl==4'b0001) result=a | b;
else if(alu_ctrl==4'b0010) result=a + b;
else if(alu_ctrl==4'b0110) result=a - b;
else if(alu_ctrl==4'b0111)
begin
    if(a[31]==1'b0 && b[31]==1'b0) begin
        if(a<b) result=32'b00000000000000000000000000000001;
        else result=32'b00000000000000000000000000000000;
    end
    else if(a[31]==1'b1 && b[31]==1'b1) begin
        if(a<b) result=32'b00000000000000000000000000000001;
        else result=32'b00000000000000000000000000000000;
    end
    else if(a[31]==1'b1 && b[31]==1'b0) begin
        result=32'b00000000000000000000000000000001;
    end
    else begin
        result=32'b00000000000000000000000000000000;
    end
end
else if(alu_ctrl==4'b0110) result=~(a | b);
else if(alu_ctrl==4'b1111) begin
    if(a==b) result=32'b00000000000000000000000000000001;
    else result=32'b00000000000000000000000000000000;
end
else ;
end

assign zero=(result==32'b0);

assign ovf=(a[31]==b[31]) & (~(a[31] ==result[31]));



endmodule
```

comparator.v

```
module comparator(equal, i0, i1);
input [31:0] i0,i1;
output equal;


assign equal=(i0==i1);
endmodule
```

control.v

```
module controller(RegDst, Jump, Branch, MemRead, MemtoReg, AluCtrl, MemWrite, ALUSrc, RegWrite, instr1, instr2);
input [5:0] instr1, instr2; //instr1=instr[31:26]; instr2=instr[5:0];
output RegDst, Jump, Branch, MemRead, MemtoReg, MemWrite, ALUSrc, RegWrite;
output [3:0] AluCtrl;

reg RegDst, Jump, Branch, MemRead, MemtoReg, MemWrite, ALUSrc, RegWrite;
reg [3:0] AluCtrl;


always @(instr1 or instr2)
begin
if(instr1==6'b0) begin
    if(instr2==6'b100000) begin
        RegDst<=1;
        Jump<=0;
        Branch<=0;
        MemRead<=0;
        MemtoReg<=0;
        AluCtrl<=4'b0010;
        MemWrite<=0;
        ALUSrc<=0;
        RegWrite<=1;
    end
    else if(instr2==6'b100010) begin
        RegDst<=1;
        Jump<=0;
        Branch<=0;
        MemRead<=0;
        MemtoReg<=0;
        AluCtrl<=4'b0110;
        MemWrite<=0;
        ALUSrc<=0;
        RegWrite<=1;
    end
    else if(instr2==6'b100100) begin
        RegDst<=1;
        Jump<=0;
        Branch<=0;
        MemRead<=0;
        MemtoReg<=0;
        AluCtrl<=4'b0000;
        MemWrite<=0;
        ALUSrc<=0;
        RegWrite<=1;
    end
    else if(instr2==6'b100101) begin
        RegDst<=1;
        Jump<=0;
        Branch<=0;
        MemRead<=0;
        MemtoReg<=0;
        AluCtrl<=4'b0001;
        MemWrite<=0;
        ALUSrc<=0;
        RegWrite<=1;
    end
    else if(instr2==6'b101010) begin
        RegDst<=1;
        Jump<=0;
        Branch<=0;
        MemRead<=0;
        MemtoReg<=0;
        AluCtrl<=4'b0111;
        MemWrite<=0;
        ALUSrc<=0;
        RegWrite<=1;
    end
    else ;
end
else begin
    if(instr1==6'b100011) begin
        RegDst<=0;
        Jump<=0;
        Branch<=0;
        MemRead<=1;
        MemtoReg<=1;
        AluCtrl<=4'b0010;
        MemWrite<=0;
        ALUSrc<=1;
        RegWrite<=1;
    end
    else if(instr1==6'b101011) begin
        RegDst<=0;
        Jump<=0;
        Branch<=0;
        MemRead<=0;
        MemtoReg<=0;
        AluCtrl<=4'b0010;
        MemWrite<=1;
        ALUSrc<=1;
        RegWrite<=0;
    end
    else if(instr1==6'b001000) begin
        RegDst<=0;
        Jump<=0;
        Branch<=0;
        MemRead<=0;
        MemtoReg<=0;
        AluCtrl<=4'b0010;
        MemWrite<=0;
        ALUSrc<=1;
        RegWrite<=1;
    end
    else if(instr1==6'b001100) begin
        RegDst<=0;
        Jump<=0;
        Branch<=0;
        MemRead<=0;
        MemtoReg<=0;
        AluCtrl<=4'b0000;
        MemWrite<=0;
        ALUSrc<=1;
        RegWrite<=1;
    end
    else if(instr1==6'b000100) begin
        RegDst<=0;
        Jump<=0;
        Branch<=1;
        MemRead<=0;
        MemtoReg<=0;
        AluCtrl<=4'b0110;
        MemWrite<=0;
        ALUSrc<=0;
        RegWrite<=0;
    end
    else if(instr1==6'b000101) begin
        RegDst<=0;
        Jump<=0;
        Branch<=1;
        MemRead<=0;
        MemtoReg<=0;
        AluCtrl<=4'b1111;
        MemWrite<=0;
        ALUSrc<=0;
        RegWrite<=0;
    end
    else if(instr1==6'b000010) begin
        RegDst<=0;
        Jump<=1;
        Branch<=0;
        MemRead<=0;
        MemtoReg<=0;
        AluCtrl<=4'b0000;
        MemWrite<=0;
        ALUSrc<=0;
        RegWrite<=0;
    end
    else ;

end

end


endmodule
```

datamem.v

```
module data_mem(Read_data, MemWrite, MemRead, Address, Write_data, clk);
input MemWrite, MemRead, clk;
input [31:0] Address, Write_data;

output [31:0] Read_data;

reg [31:0] m [0:20000];// A modelled memory. The size is relatively small.

reg [31:0] i3;
initial begin
    for(i3=0;i3<=20000;i3=i3+1)
    begin
    m[i3]=0;
    end
end


wire [31:0] Read_data;


assign Read_data=m[Address>>2];


always @(negedge clk)
begin
//if(MemRead==1'b1) begin
//    Read_data<=m[Address>>2];
//end
if(MemWrite==1'b1) begin
    m[Address>>2]<=Write_data;
end
end
endmodule
```

EX_MEM_regf.v

```
module EX_MEM_regf(MemtoReg_o, RegWrite_o, Branch_o, MemWrite_o,
MemRead_o, aluo_o, rego_o, I5_o,
MemtoReg_i, RegWrite_i, Branch_i, MemWrite_i,
MemRead_i, aluo_i, rego_i, I5_i, clk);
input MemtoReg_i, RegWrite_i, Branch_i, MemWrite_i, MemRead_i, clk;
input [31:0] aluo_i, rego_i, I5_i;
output MemtoReg_o, RegWrite_o, Branch_o, MemWrite_o, MemRead_o;
output [31:0] aluo_o, rego_o, I5_o;

reg MemtoReg_o, RegWrite_o, Branch_o, MemWrite_o, MemRead_o;
reg [31:0] aluo_o, rego_o, I5_o;

always @(posedge clk)
begin
MemtoReg_o=MemtoReg_i;
RegWrite_o=RegWrite_i;
Branch_o=Branch_i;
MemWrite_o=MemWrite_i;
MemRead_o=MemRead_i;
aluo_o=aluo_i;
rego_o=rego_i;
I5_o=I5_i;

end

endmodule
```

extender.v

```
module extender(out, i);
input [15:0] i;
output [31:0] out;


reg [31:0] out;

always @(i) begin
    if(i[15]==1'b0) out={16'b0000000000000000, i};
    else out={16'b1111111111111111, i};
end

endmodule
```

forwarding.v

```
module forwarding(sel1, sel2, ID_EX_rs, ID_EX_rt, EX_MEM_rd, EX_MEM_regwrite, MEM_WB_rd, MEM_WB_regwrite);
input [4:0] ID_EX_rs, ID_EX_rt, EX_MEM_rd, MEM_WB_rd;
input EX_MEM_regwrite, MEM_WB_regwrite;
output [1:0] sel1,sel2;

reg [1:0] sel1, sel2;

always @(ID_EX_rs or ID_EX_rt or EX_MEM_rd or EX_MEM_regwrite or MEM_WB_rd or MEM_WB_regwrite)
begin
if(EX_MEM_regwrite==1'b1 && (EX_MEM_rd!=1'b0) && (EX_MEM_rd==ID_EX_rs)) sel1=2'b10;
else if(MEM_WB_regwrite==1'b1 && (MEM_WB_rd !=1'b0) && (MEM_WB_rd == ID_EX_rs)) sel1=2'b01;
else sel1=2'b00;


if(EX_MEM_regwrite==1 && (EX_MEM_rd!=0) && (EX_MEM_rd==ID_EX_rt)) sel2=2'b10;
else if(MEM_WB_regwrite==1'b1 && (MEM_WB_rd !=1'b0) && (MEM_WB_rd == ID_EX_rt)) sel2=2'b01;
else sel2=2'b00;

end

endmodule
```

Hazard_detection.v

```
module hazard_detection(src, enable_pc, IF_ID_enable, IF_ID_flush,
 lw_hazard, ID_EX_memread, memread_DM, inst ,
  ID_EX_rt, EX_MEM_rd, ID_EX_regwrite, EX_MEM_regwrite, equal, clk);
input ID_EX_memread, equal, memread_DM, ID_EX_regwrite, EX_MEM_regwrite, clk;
input [31:0] inst;
input [4:0] ID_EX_rt, EX_MEM_rd;
output [1:0] src;
output enable_pc, IF_ID_enable, IF_ID_flush, lw_hazard;
reg [1:0] src;
reg enable_pc, IF_ID_enable, lw_hazard, IF_ID_flush;
//detects problems with lw, beq, bne and j.

initial begin
enable_pc=1'b0;
src=2'b00;
end

always @(ID_EX_memread or equal or memread_DM or ID_EX_regwrite or
EX_MEM_regwrite or inst or ID_EX_rt or EX_MEM_rd or clk)
begin
    if (((ID_EX_regwrite==1'b1 && (ID_EX_rt==inst[25:21] ||ID_EX_rt==inst[20:16]))||
    (EX_MEM_regwrite==1'b1 && (EX_MEM_rd==inst[25:21] || EX_MEM_rd==inst[20:16]))) && (
    inst[31:26]==6'b000100 || inst[31:26]==6'b000101)) begin
        IF_ID_enable=1'b0;
        lw_hazard=1'b1;
        src=2'b00;
        enable_pc=1'b0;
        IF_ID_flush=1'b0;
    end
    else if(ID_EX_memread==1'b1  && ( ID_EX_rt==inst[25:21] || ID_EX_rt==inst[20:16] ))  //lw
    begin
        IF_ID_enable=1'b0;
        lw_hazard=1'b1;
        src=2'b00;
        enable_pc=1'b0;
        IF_ID_flush=1'b0;
    end
   else if(inst[31:26]==6'b000100 && equal==1'b1) // beq
   begin
        IF_ID_flush=1'b1;
        src=2'b10;
        enable_pc=1'b1;
        IF_ID_enable=1'b1;
        lw_hazard=1'b0;
   end
   else if(inst[31:26]==6'b000101 && equal==1'b0)// bne
   begin
        IF_ID_flush=1'b1;
        src=2'b10;
        enable_pc=1'b1;
        IF_ID_enable=1'b1;
        lw_hazard=1'b0;
   end
   else if(inst[31:26]==6'b000010) //j
   begin
        IF_ID_flush=1'b1;
        src=2'b01;
        enable_pc=1'b1;
        IF_ID_enable=1'b1;
        lw_hazard=1'b0;
   end
```

```
    else begin
         IF_ID_flush=1'b0;
         src=2'b00;
         enable_pc=1'b1;
         IF_ID_enable=1'b1;
         lw_hazard=1'b0;
    end

end

endmodule
```

ID_EX_regf.v

```
module ID_EX_regf(MemtoReg_o, RegWrite_o, Branch_o, MemWrite_o,
MemRead_o, RegDst_o, AluControl_o, ALUSrc_o, read1_o, read2_o,
 extension_o, I1_o, I2_o, I3_o,
 MemtoReg_i, RegWrite_i, Branch_i, MemWrite_i, MemRead_i, RegDst_i,
 AluControl_i, ALUSrc_i, read1_i, read2_i, extension_i, I1_i, I2_i, I3_i, clk);

input MemtoReg_i, RegWrite_i, Branch_i, MemWrite_i, MemRead_i, RegDst_i, ALUSrc_i, clk;
input [3:0] AluControl_i;
input [4:0] I1_i, I2_i, I3_i;
input [31:0] read1_i, read2_i, extension_i;
output MemtoReg_o, RegWrite_o, Branch_o, MemWrite_o,
MemRead_o, RegDst_o, ALUSrc_o;
output [3:0] AluControl_o;
output [4:0] I1_o, I2_o, I3_o;
output [31:0] read1_o, read2_o, extension_o;


reg MemtoReg_o, RegWrite_o, Branch_o, MemWrite_o,
MemRead_o, RegDst_o, ALUSrc_o;
reg [3:0] AluControl_o;
reg [4:0] I1_o, I2_o, I3_o;
reg [31:0] read1_o, read2_o, extension_o;



always @(posedge clk)
begin
    MemtoReg_o=MemtoReg_i;
    RegWrite_o=RegWrite_i;
    Branch_o=Branch_i;
    MemWrite_o=MemWrite_i;
    MemRead_o=MemRead_i;
    RegDst_o=RegDst_i;
    ALUSrc_o=ALUSrc_i;
    AluControl_o=AluControl_i;
    I1_o=I1_i;
    I2_o=I2_i;
    I3_o=I3_i;
    read1_o=read1_i;
    read2_o=read2_i;
    extension_o=extension_i;
end

endmodule
```

IF_ID_regf.v

```
module IF_ID_regf(PC_out,inst_out,PC_in, inst_in, write_en, flush, clk);
input write_en, flush, clk;
input [31:0] PC_in, inst_in;
output [31:0] PC_out, inst_out;


reg [31:0] PC_out, inst_out;
always @(posedge clk)
begin
    if(flush==1'b1) inst_out=32'b00000000000000000000000000000000;
    else if(write_en==1'b0) ;
    else begin

    PC_out=PC_in;
    inst_out=inst_in;

    end
end



endmodule
```

instmem.v

```
module inst_mem(inst, pc);
input [31:0] pc;

output [31:0] inst;

reg [31:0] m [0:19999];// A modelled memory. The size is relatively small.
reg [7:0] m1 [0:80000];
reg [31:0] i1, i2;


initial begin
for(i2=0; i2<=80000;i2 = i2 + 1) begin
    m1[i2]=0;
end
//$readmemb("InstructionMem_for_P2_Demo_bonus.txt",m1);
$readmemb("InstructionMem_for_P2_Demo.txt",m1);
for(i1=0; i1<20000; i1=i1+1) begin
    m[i1]<={m1[i1*4],m1[i1*4+1],m1[i1*4+2],m1[i1*4+3]};
end
end


wire [31:0] inst;

assign inst=m[pc>>2];


endmodule
```

MEM_WB_regf.v

```
module MEM_WB_regf(MemtoReg_o, RegWrite_o, Mem_o, aluo_o, I6_o,
 MemtoReg_i, RegWrite_i, Mem_i, aluo_i, I6_i, clk);
input MemtoReg_i, RegWrite_i, clk;
input [31:0] Mem_i, aluo_i, I6_i;
output MemtoReg_o, RegWrite_o;
output [31:0] Mem_o, aluo_o, I6_o;

reg MemtoReg_o, RegWrite_o;
reg [31:0]  Mem_o, aluo_o, I6_o;

always @(posedge clk)
begin
    MemtoReg_o=MemtoReg_i;
    RegWrite_o=RegWrite_i;
    Mem_o=Mem_i;
    aluo_o=aluo_i;
    I6_o=I6_i;
end


endmodule
```

mux21.v

```
module mux21(out,i0,i1,sel);
parameter width=32;
input sel;
input [width-1:0] i0,i1;
output [width-1:0] out;
reg [width-1:0] out;



always @(i0 or i1 or sel) begin
    if(sel==1'b1) out<=i1;
    else out<=i0;
end

endmodule
```

mux41.v

```
module mux41(out,i0,i1,i2,i3,sel);
parameter width=32;
input [width-1:0] i0,i1,i2,i3;
input [1:0] sel;
output [width-1:0] out;
```

```
reg [width-1:0] out;



always @(i0 or i1 or i2 or i3 or sel) begin
    if(sel==2'b00) out<=i0;
    else if (sel==2'b01) out<=i1;
    else if (sel==2'b10) out<=i2;
    else out<=i3;
end

endmodule
```

reg.v

```
module register(o,i,clk,enable);
input [31:0] i;
input enable;
input clk;
output [31:0] o;

reg [31:0] o;



initial begin
    o=0;
end

always @(posedge clk)
begin
    if(enable==1'b1) begin
        o=i;
    end
    else begin
        ;
    end
end

endmodule
```

regfile.v

```
module register_file(s0, s1,
         s2, s3, s4, s5, s6, s7, t0, t1, t2, t3, t4, t5, t6, t7, t8, t9, Read_data1, Read_data2, RegWrite,
Read_register1, Read_register2, Write_register, Write_data, clk);
input RegWrite,clk;
input [4:0] Read_register1, Read_register2, Write_register;
input [31:0] Write_data;

output [31:0] Read_data1, Read_data2;

output [31:0]  s0, s1, s2, s3, s4, s5, s6, s7, t0, t1, t2, t3, t4, t5, t6, t7, t8, t9;

reg [31:0] regf [0:31];


wire [31:0] Read_data1, Read_data2;
wire [31:0]  s0, s1, s2, s3, s4, s5, s6, s7, t0, t1, t2, t3, t4, t5, t6, t7, t8, t9;


initial begin
    regf[0] = 0;
    regf[1] = 0;
    regf[2] = 0;
    regf[3] = 0;
    regf[4] = 0;
    regf[5] = 0;
    regf[6] = 0;
    regf[7] = 0;
    regf[8] = 0;
    regf[9] = 0;
    regf[10] = 0;
    regf[11] = 0;
    regf[12] = 0;
    regf[13] = 0;
    regf[14] = 0;
    regf[15] = 0;
    regf[16] = 0;
    regf[17] = 0;
    regf[18] = 0;
    regf[19] = 0;
    regf[20] = 0;
    regf[21] = 0;
    regf[22] = 0;
    regf[23] = 0;
    regf[24] = 0;
    regf[25] = 0;
    regf[26] = 0;
    regf[27] = 0;
    regf[28] = 0;
    regf[29] = 0;
    regf[30] = 0;
    regf[31] = 0;
end

    assign s0=regf[16];
    assign s1=regf[17];
    assign s2=regf[18];
    assign s3=regf[19];
    assign s4=regf[20];
    assign s5=regf[21];
    assign s6=regf[22];
    assign s7=regf[23];


    assign t0=regf[8];
    assign t1=regf[9];
    assign t2=regf[10];
    assign t3=regf[11];
    assign t4=regf[12];
    assign t5=regf[13];
    assign t6=regf[14];
    assign t7=regf[15];
    assign t8=regf[24];
    assign t9=regf[25];

    assign Read_data1=regf[Read_register1];
    assign Read_data2=regf[Read_register2];


always @(negedge clk)
begin
regf[0]=0;
if(RegWrite==1'b1 && Write_register != 0 ) begin
    regf[Write_register]=Write_data;
end
else begin
    ;
end




end


endmodule
```

shifter.v

```
module shifter1(out, i);
input [25:0] i;
output [27:0] out;

reg [27:0] out;

always @(i)
begin
    out={i, 2'b00};
end
endmodule
```

shifter2.v

```
module shifter2(out, i);
input [31:0] i;
output [31:0] out;

reg [31:0] out;

always @(i)
begin
    out={i[29:0], 2'b00};
end
endmodule
```

The following codes are for FPGA simulation.

clock_divider_1.v

```
module clock_divider_1(clock1,clock,reset);//clock divider 100M->500
input clock, reset;
output clock1;


reg [17:0] Q;
reg clock1;

always @(posedge clock or posedge reset)
begin
    if (reset==1'b1) Q<=18'b000000000000000000;
    else if (Q==18'b110000110100111111) Q<=18'b000000000000000000;
    else  Q<=Q+1;

    clock1=(Q==18'b110000110100111111);
end
endmodule
```

clock_divider_2.v

```
module clock_divider_2(clock2,clock1,reset);//clock divider: 500->1
input clock1, reset;
output clock2;


reg [8:0] Q;
reg clock2;

always @(posedge clock1 or posedge reset)
begin
    if (reset==1'b1) Q<=9'b000000000;
    else if (Q==9'b111110011) Q<=9'b000000000;
    else  Q<=Q+1;

    clock2=(Q==9'b111110011);
end
endmodule
```

Hazard_detection.v

```
module hazard_detection(src, enable_pc, IF_ID_enable, IF_ID_flush,
 lw_hazard, ID_EX_memread, memread_DM, inst ,
  ID_EX_rt, EX_MEM_rd, ID_EX_regwrite, EX_MEM_regwrite, equal, clk, reset);
input ID_EX_memread, equal, memread_DM, ID_EX_regwrite, EX_MEM_regwrite, clk, reset;
input [31:0] inst;
input [4:0] ID_EX_rt, EX_MEM_rd;
output [1:0] src;
output enable_pc, IF_ID_enable, IF_ID_flush, lw_hazard;
reg [1:0] src;
reg enable_pc, IF_ID_enable, lw_hazard, IF_ID_flush;
//detects problems with lw, beq, bne and j.

always @(ID_EX_memread or equal or memread_DM or ID_EX_regwrite or
EX_MEM_regwrite or inst or ID_EX_rt or EX_MEM_rd or clk or reset)
begin
    if(reset==1'b1) begin
    enable_pc=1'b1;
    src=2'b00;
    end


    if (((ID_EX_regwrite==1'b1 && (ID_EX_rt==inst[25:21] ||ID_EX_rt==inst[20:16]))||
    (EX_MEM_regwrite==1'b1 && (EX_MEM_rd==inst[25:21] || EX_MEM_rd==inst[20:16]))) && (
    inst[31:26]==6'b000100 || inst[31:26]==6'b000101)) begin
        IF_ID_enable=1'b0;
        lw_hazard=1'b1;
        src=2'b00;
        enable_pc=1'b0;
        IF_ID_flush=1'b0;
    end
    else if(ID_EX_memread==1'b1  && ( ID_EX_rt==inst[25:21] || ID_EX_rt==inst[20:16] ))  //lw
    begin
        IF_ID_enable=1'b0;
        lw_hazard=1'b1;
        src=2'b00;
        enable_pc=1'b0;
        IF_ID_flush=1'b0;
     end
    else if(inst[31:26]==6'b000100 && equal==1'b1) // beq
    begin
        IF_ID_flush=1'b1;
        src=2'b10;
        enable_pc=1'b1;
        IF_ID_enable=1'b1;
        lw_hazard=1'b0;
    end
    else if(inst[31:26]==6'b000101 && equal==1'b0)// bne
    begin
        IF_ID_flush=1'b1;
        src=2'b10;
        enable_pc=1'b1;
        IF_ID_enable=1'b1;
        lw_hazard=1'b0;
    end
    else if(inst[31:26]==6'b000010) //j
    begin
        IF_ID_flush=1'b1;
        src=2'b01;
        enable_pc=1'b1;
        IF_ID_enable=1'b1;
        lw_hazard=1'b0;
    end
    else begin
        IF_ID_flush=1'b0;
        src=2'b00;
        enable_pc=1'b1;
        IF_ID_enable=1'b1;
        lw_hazard=1'b0;
    end

end

endmodule
```

instmem.v

```
module inst_mem(inst, pc, reset);
input [31:0] pc;
input reset;

output [31:0] inst;

reg [31:0] m [0:150];// A modelled memory. The size is relatively small.


wire [31:0] inst;

assign inst=m[pc>>2];


always @(posedge reset) begin
if(reset==1'b1) begin
//m[0]=32'b00100000000010000000000000100000; //addi $t0, $zero, 0x20
//m[1]=32'b00100000000010010000000000110111;//addi $t1, $zero, 0x37
//m[2]=32'b00000001000010011000000000100100;//and $s0, $t0, $t1
//m[3]=32'b00000001000010011000000000100101; //or $s0, $t0, $t1
//m[4]=32'b10101100000100000000000000000100; //sw $s0, 4($zero)
//m[5]=32'b10101100000010000000000000001000; //sw $t0, 8($zero)
//m[6]=32'b00000001000010011000100000100000; //add $s1, $t0, $t1
//m[7]=32'b00000001000010011001000000100010;//sub $s2, $t0, $t1
//m[8]=32'b00100000000010000000000000100000; //addi $t0, $zero, 0x20
//m[9]=32'b00100000000010000000000000100000; //addi $t0, $zero, 0x20
//m[10]=32'b00100000000010000000000000100000; //addi $t0, $zero, 0x20
//m[11]=32'b00010010001100100000000000010010; //beq $s1, $s2, error0
//m[12]=32'b10001100000100010000000000000100; //lw $s1, 4($zero)
//m[13]=32'b00110010001100100000000001001000; //andi $s2, $s1, 0x48
//m[14]=32'b00100000000010000000000000100000; //addi $t0, $zero, 0x20
//m[15]=32'b00100000000010000000000000100000; //addi $t0, $zero, 0x20
//m[16]=32'b00100000000010000000000000100000; //addi $t0, $zero, 0x20
//m[17]=32'b00010010001100100000000000001111; //beq $s1, $s2, error1
//m[18]=32'b10001100000100110000000000001000; //lw $s3, 8($zero)
//m[19]=32'b00100000000010000000000000100000; //addi $t0, $zero, 0x20
//m[20]=32'b00100000000010000000000000100000; //addi $t0, $zero, 0x20
//m[21]=32'b00100000000010000000000000100000; //addi $t0, $zero, 0x20
//m[22]=32'b00010010000100110000000000001101; //beq $s0, $s3, error2
//m[23]=32'b00000010010100011010000000101010; //slt $s4, $s2, $s1 (Last)
//m[24]=32'b00100000000010000000000000100000; //addi $t0, $zero, 0x20
//m[25]=32'b00100000000010000000000000100000; //addi $t0, $zero, 0x20
//m[26]=32'b00100000000010000000000000100000; //addi $t0, $zero, 0x20
```

```
//m[27]=32'b00010010100000000000000000001111; //beq $s4, $0, EXIT
//m[28]=32'b00000010001000001001000000100000; //add $s2, $s1, $0
//m[29]=32'b00001000000000000000000000010111; //j Last
//m[30]=32'b00100000000010000000000000000000; //addi $t0, $0, 0(error0)
//m[31]=32'b00100000000010010000000000000000; //addi $t1, $0, 0
//m[32]=32'b00001000000000000000000000111111; //j EXIT
//m[33]=32'b00100000000010000000000000000001; //addi $t0, $0, 1(error1)
//m[34]=32'b00100000000010010000000000000001; //addi $t1, $0, 1
//m[35]=32'b00001000000000000000000000111111; //j EXIT
//m[36]=32'b00100000000010000000000000000010; //addi $t0, $0, 2(error2)
//m[37]=32'b00100000000010010000000000000010; //addi $t1, $0, 2
//m[38]=32'b00001000000000000000000000111111; //j EXIT
//m[39]=32'b00100000000010000000000000000011; //addi $t0, $0, 3(error3)
//m[40]=32'b00100000000010010000000000000011; //addi $t1, $0, 3
//m[41]=32'b00001000000000000000000000111111; //j EXIT
//m[42] = 0;
//m[43] = 0;
//m[44] = 0;
//m[45] = 0;
//m[46] = 0;
//m[47] = 0;
//m[48] = 0;
//m[49] = 0;
//m[50] = 0;
//m[51] = 0;
//m[52] = 0;
//m[53] = 0;
//m[54] = 0;
//m[55] = 0;
//m[56] = 0;
//m[57] = 0;
//m[58] = 0;
//m[59] = 0;
//m[60] = 0;
//m[61] = 0;
//m[62] = 0;
//m[63] = 0;
//m[64] = 0;
//m[65] = 0;
//m[66] = 0;
//m[67] = 0;
//m[68] = 0;
//m[69] = 0;
//m[70] = 0;
//m[71] = 0;
//m[72] = 0;
//m[73] = 0;
//m[74] = 0;
//m[75] = 0;
//m[76] = 0;
//m[77] = 0;
//m[78] = 0;
//m[79] = 0;
//m[80] = 0;
//m[81] = 0;
//m[82] = 0;
//m[83] = 0;
//m[84] = 0;
//m[85] = 0;
//m[86] = 0;
//m[87] = 0;
//m[88] = 0;
//m[89] = 0;
//m[90] = 0;
//m[91] = 0;
//m[92] = 0;
//m[93] = 0;
//m[94] = 0;
//m[95] = 0;
//m[96] = 0;
//m[97] = 0;
//m[98] = 0;
//m[99] = 0;
//m[100] = 0;
//m[101] = 0;
//m[102] = 0;
//m[103] = 0;
//m[104] = 0;
//m[105] = 0;
//m[106] = 0;
//m[107] = 0;
//m[108] = 0;
//m[109] = 0;
//m[110] = 0;
//m[111] = 0;
//m[112] = 0;
//m[113] = 0;
```

```
    //m[114] = 0;
    //m[115] = 0;
    //m[116] = 0;
    //m[117] = 0;
    //m[118] = 0;
    //m[119] = 0;
    //m[120] = 0;
    //m[121] = 0;
    //m[122] = 0;
    //m[123] = 0;
    //m[124] = 0;
    //m[125] = 0;
    //m[126] = 0;
    //m[127] = 0;
    //m[128] = 0;
    //m[129] = 0;
    //m[130] = 0;
    //m[131] = 0;
    //m[132] = 0;
    //m[133] = 0;
    //m[134] = 0;
    //m[135] = 0;
    //m[136] = 0;
    //m[137] = 0;
    //m[138] = 0;
    //m[139] = 0;
    //m[140] = 0;
    //m[141] = 0;
    //m[142] = 0;
    //m[143] = 0;
    //m[144] = 0;
    //m[145] = 0;
    //m[146] = 0;
    //m[147] = 0;
    //m[148] = 0;
    //m[149] = 0;
    //m[150] = 0;

    m[0]=32'b00100000000010000000000000100000; //addi $t0, $zero, 0x20
    m[1]=32'b00100000000010010000000000110111;//addi $t1, $zero, 0x37
    m[2]=32'b00000001000010011000000000100100;//and $s0, $t0, $t1
    m[3]=32'b00000001000010011000000000100101; //or $s0, $t0, $t1
    m[4]=32'b10101100000100000000000000000100; //sw $s0, 4($zero)
    m[5]=32'b10101100000010000000000000001000; //sw $t0, 8($zero)
    m[6]=32'b00000001000010011000100000100000; //add $s1, $t0, $t1
    m[7]=32'b00000001000010011001000000100010;//sub $s2, $t0, $t1
    m[8]=32'b00010010001100100000000000001001; //beq $s1, $s2, error0
    m[9]=32'b10001100000100010000000000000100; //lw $s1, 4($zero)
    m[10]=32'b00110010001100100000000001001000; //andi $s2, $s1, 0x48
    m[11]=32'b00010010001100100000000000001001; //beq $s1, $s2, error1
    m[12]=32'b10001100000100110000000000001000; //lw $s3, 8($zero)
    m[13]=32'b00010010000100110000000000001010; //beq $s0, $s3, error2
    m[14]=32'b00000010010100011010000000101010; //slt $s4, $s2, $s1 (Last)
    m[15]=32'b00010010100000000000000000001111; //beq $s4, $0, EXIT
    m[16]=32'b00000010001000001001000000100000; //add $s2, $s1, $0
    m[17]=32'b00001000000000000000000000001110; //j Last
    m[18]=32'b00100000000010000000000000000000; //addi $t0, $0, 0(error0)
    m[19]=32'b00100000000010010000000000000000; //addi $t1, $0, 0
    m[20]=32'b00001000000000000000000000011111; //j EXIT
    m[21]=32'b00100000000010000000000000000001; //addi $t0, $0, 1(error1)
    m[22]=32'b00100000000010010000000000000001; //addi $t1, $0, 1
    m[23]=32'b00001000000000000000000000011111; //j EXIT
    m[24]=32'b00100000000010000000000000000010; //addi $t0, $0, 2(error2)
    m[25]=32'b00100000000010010000000000000010; //addi $t1, $0, 2
    m[26]=32'b00001000000000000000000000011111; //j EXIT
    m[27]=32'b00100000000010000000000000000011; //addi $t0, $0, 3(error3)
    m[28]=32'b00100000000010010000000000000011; //addi $t1, $0, 3
    m[29]=32'b00001000000000000000000000011111; //j EXIT
    m[30] = 0;
    m[31] = 0;
    m[32] = 0;
    m[33] = 0;
    m[34] = 0;
    m[35] = 0;
    m[36] = 0;
    m[37] = 0;
    m[38] = 0;
    m[39] = 0;
    m[40] = 0;
    m[41] = 0;
    m[42] = 0;
    m[43] = 0;
    m[44] = 0;
    m[45] = 0;
    m[46] = 0;
    m[47] = 0;
    m[48] = 0;
```

```
  m[49] = 0;
  m[50] = 0;
  m[51] = 0;
  m[52] = 0;
  m[53] = 0;
  m[54] = 0;
  m[55] = 0;
  m[56] = 0;
  m[57] = 0;
  m[58] = 0;
  m[59] = 0;
  m[60] = 0;
  m[61] = 0;
  m[62] = 0;
  m[63] = 0;
  m[64] = 0;
  m[65] = 0;
  m[66] = 0;
  m[67] = 0;
  m[68] = 0;
  m[69] = 0;
  m[70] = 0;
  m[71] = 0;
  m[72] = 0;
  m[73] = 0;
  m[74] = 0;
  m[75] = 0;
  m[76] = 0;
  m[77] = 0;
  m[78] = 0;
  m[79] = 0;
  m[80] = 0;
  m[81] = 0;
  m[82] = 0;
  m[83] = 0;
  m[84] = 0;
  m[85] = 0;
  m[86] = 0;
  m[87] = 0;
  m[88] = 0;
  m[89] = 0;
  m[90] = 0;
  m[91] = 0;
  m[92] = 0;
  m[93] = 0;
  m[94] = 0;
  m[95] = 0;
  m[96] = 0;
  m[97] = 0;
  m[98] = 0;
  m[99] = 0;
  m[100] = 0;
  m[101] = 0;
  m[102] = 0;
  m[103] = 0;
  m[104] = 0;
  m[105] = 0;
  m[106] = 0;
  m[107] = 0;
  m[108] = 0;
  m[109] = 0;
  m[110] = 0;
  m[111] = 0;
  m[112] = 0;
  m[113] = 0;
  m[114] = 0;
  m[115] = 0;
  m[116] = 0;
  m[117] = 0;
  m[118] = 0;
  m[119] = 0;
  m[120] = 0;
  m[121] = 0;
  m[122] = 0;
  m[123] = 0;
  m[124] = 0;
  m[125] = 0;
  m[126] = 0;
  m[127] = 0;
  m[128] = 0;
  m[129] = 0;
  m[130] = 0;
  m[131] = 0;
  m[132] = 0;
  m[133] = 0;
  m[134] = 0;
  m[135] = 0;
```

```
m[136] = 0;
m[137] = 0;
m[138] = 0;
m[139] = 0;
m[140] = 0;
m[141] = 0;
m[142] = 0;
m[143] = 0;
m[144] = 0;
m[145] = 0;
m[146] = 0;
m[147] = 0;
m[148] = 0;
m[149] = 0;
m[150] = 0;



end
end


endmodule
```

pipeline_2.v

```
module pipeline_implementation(A, C, reg_addr, real_clk, clk, reset);
input real_clk, clk, reset;
input [5:0] reg_addr;
output [3:0] A; //AN3 AN2 AN1 AN0
output [6:0] C;//CA CB CC CD... CG
wire [31:0] pc, zero, at, v0, v1, a0, a1, a2, a3, s0, s1,
         s2, s3, s4, s5, s6, s7, t0, t1, t2, t3, t4, t5, t6, t7, t8, t9, k0, k1, gp, sp, fp, ra;

reg [4:0] C3,C2,C1,C0;//value

wire [6:0] C3_o, C2_o, C1_o, C0_o;

clock_divider_1 c1(clock1,real_clk,reset);
ring_counter rc1(A, clock1, reset);


pipeline_processor pipeline_impl1(zero, at, v0, v1, a0, a1, a2, a3, s0, s1,
         s2, s3, s4, s5, s6, s7, t0, t1, t2, t3, t4, t5, t6, t7, t8, t9, k0, k1, gp, sp, fp, ra, pc,
          clk, reset);


reg [6:0] C;
always @(reg_addr)
begin
    if(reg_addr==6'b000000) begin
        C3<=zero[15:12];
        C2<=zero[11:8];
        C1<=zero[7:4];
        C0<=zero[3:0];
    end
    else if(reg_addr==6'b000001) begin
        C3<=at[15:12];
        C2<=at[11:8];
        C1<=at[7:4];
        C0<=at[3:0];
    end
    else if(reg_addr==6'b000010) begin
        C3<=v0[15:12];
        C2<=v0[11:8];
        C1<=v0[7:4];
        C0<=v0[3:0];
    end
    else if(reg_addr==6'b000011) begin
        C3<=v1[15:12];
        C2<=v1[11:8];
        C1<=v1[7:4];
        C0<=v1[3:0];
    end
    else if(reg_addr==6'b000100) begin
        C3<=a0[15:12];
        C2<=a0[11:8];
        C1<=a0[7:4];
        C0<=a0[3:0];
    end
    else if(reg_addr==6'b000101) begin
        C3<=a1[15:12];
        C2<=a1[11:8];
        C1<=a1[7:4];
        C0<=a1[3:0];
```

```
      end
      else if(reg_addr==6'b000110) begin
          C3<=a2[15:12];
          C2<=a2[11:8];
          C1<=a2[7:4];
          C0<=a2[3:0];
      end
      else if(reg_addr==6'b000111) begin
          C3<=a3[15:12];
          C2<=a3[11:8];
          C1<=a3[7:4];
          C0<=a3[3:0];
      end
      else if(reg_addr==6'b001000) begin
          C3<=t0[15:12];
          C2<=t0[11:8];
          C1<=t0[7:4];
          C0<=t0[3:0];
      end
      else if(reg_addr==6'b001001) begin
          C3<=t1[15:12];
          C2<=t1[11:8];
          C1<=t1[7:4];
          C0<=t1[3:0];
      end
      else if(reg_addr==6'b001010) begin
          C3<=t2[15:12];
          C2<=t2[11:8];
          C1<=t2[7:4];
          C0<=t2[3:0];
      end
      else if(reg_addr==6'b001011) begin
          C3<=t3[15:12];
          C2<=t3[11:8];
          C1<=t3[7:4];
          C0<=t3[3:0];
      end
      else if(reg_addr==6'b001100) begin
          C3<=t4[15:12];
          C2<=t4[11:8];
          C1<=t4[7:4];
          C0<=t4[3:0];
      end
      else if(reg_addr==6'b001101) begin
          C3<=t5[15:12];
          C2<=t5[11:8];
          C1<=t5[7:4];
          C0<=t5[3:0];
      end
      else if(reg_addr==6'b001110) begin
          C3<=t6[15:12];
          C2<=t6[11:8];
          C1<=t6[7:4];
          C0<=t6[3:0];
      end
      else if(reg_addr==6'b001111) begin
          C3<=t7[15:12];
          C2<=t7[11:8];
          C1<=t7[7:4];
          C0<=t7[3:0];
      end
      else if(reg_addr==6'b010000) begin
          C3<=s0[15:12];
          C2<=s0[11:8];
          C1<=s0[7:4];
          C0<=s0[3:0];
      end
      else if(reg_addr==6'b010001) begin
          C3<=s1[15:12];
          C2<=s1[11:8];
          C1<=s1[7:4];
          C0<=s1[3:0];
      end
      else if(reg_addr==6'b010010) begin
          C3<=s2[15:12];
          C2<=s2[11:8];
          C1<=s2[7:4];
          C0<=s2[3:0];
      end
      else if(reg_addr==6'b010011) begin
          C3<=s3[15:12];
          C2<=s3[11:8];
          C1<=s3[7:4];
          C0<=s3[3:0];
      end
      else if(reg_addr==6'b010100) begin
          C3<=s4[15:12];
```

```
        C2<=s4[11:8];
        C1<=s4[7:4];
        C0<=s4[3:0];
    end
    else if(reg_addr==6'b010101) begin
        C3<=s5[15:12];
        C2<=s5[11:8];
        C1<=s5[7:4];
        C0<=s5[3:0];
    end
    else if(reg_addr==6'b010110) begin
        C3<=s6[15:12];
        C2<=s6[11:8];
        C1<=s6[7:4];
        C0<=s6[3:0];
    end
    else if(reg_addr==6'b010111) begin
        C3<=s7[15:12];
        C2<=s7[11:8];
        C1<=s7[7:4];
        C0<=s7[3:0];
    end
    else if(reg_addr==6'b011000) begin
        C3<=t8[15:12];
        C2<=t8[11:8];
        C1<=t8[7:4];
        C0<=t8[3:0];
    end
    else if(reg_addr==6'b011001) begin
        C3<=t9[15:12];
        C2<=t9[11:8];
        C1<=t9[7:4];
        C0<=t9[3:0];
    end
    else if(reg_addr==6'b011010) begin
        C3<=k0[15:12];
        C2<=k0[11:8];
        C1<=k0[7:4];
        C0<=k0[3:0];
    end
    else if(reg_addr==6'b011011) begin
        C3<=k1[15:12];
        C2<=k1[11:8];
        C1<=k1[7:4];
        C0<=k1[3:0];
    end
    else if(reg_addr==6'b011100) begin
        C3<=gp[15:12];
        C2<=gp[11:8];
        C1<=gp[7:4];
        C0<=gp[3:0];
    end
    else if(reg_addr==6'b011101) begin
        C3<=sp[15:12];
        C2<=sp[11:8];
        C1<=sp[7:4];
        C0<=sp[3:0];
    end
    else if(reg_addr==6'b011110) begin
        C3<=fp[15:12];
        C2<=fp[11:8];
        C1<=fp[7:4];
        C0<=fp[3:0];
    end
    else if(reg_addr==6'b011111) begin
        C3<=ra[15:12];
        C2<=ra[11:8];
        C1<=ra[7:4];
        C0<=ra[3:0];
    end
    else if(reg_addr==6'b100000) begin
        C3<=pc[15:12];
        C2<=pc[11:8];
        C1<=pc[7:4];
        C0<=pc[3:0];
    end



end



ssd_driver_new ssd1(C3_o, C3);
ssd_driver_new ssd2(C2_o, C2);
ssd_driver_new ssd3(C1_o, C1);
ssd_driver_new ssd4(C0_o, C0);
```

```
always @(A)
begin
if(A==4'b0111) C<=C3_o;
else if (A==4'b1011) C<=C2_o;
else if (A==4'b1101) C<=C1_o;
else C<=C0_o;
end

endmodule
```

reg.v

```
module register(o,i,clk,enable, reset);
input [31:0] i;
input enable;
input clk, reset;
output [31:0] o;

reg [31:0] o;




always @(posedge clk or posedge reset)
begin
    if(reset==1'b1) begin
        o=0;
    end
    else if(enable==1'b1) begin
        o=i;
    end
    else begin
        ;
    end
end

endmodule
```

ring_counter.v

```
module ring_counter(Q, clock, reset);
input clock, reset;
output [3:0] Q;

reg [3:0] Q;
always @(posedge clock or posedge reset)
begin
    if (reset==1'b1) Q<=4'b0111;
    else if (Q==4'b0111) Q<=4'b1011;
    else if (Q==4'b1011) Q<=4'b1101;
    else if (Q==4'b1101) Q<=4'b1110;
    else Q<=4'b0111;
end

endmodule
```

ssd_driver_modify.v

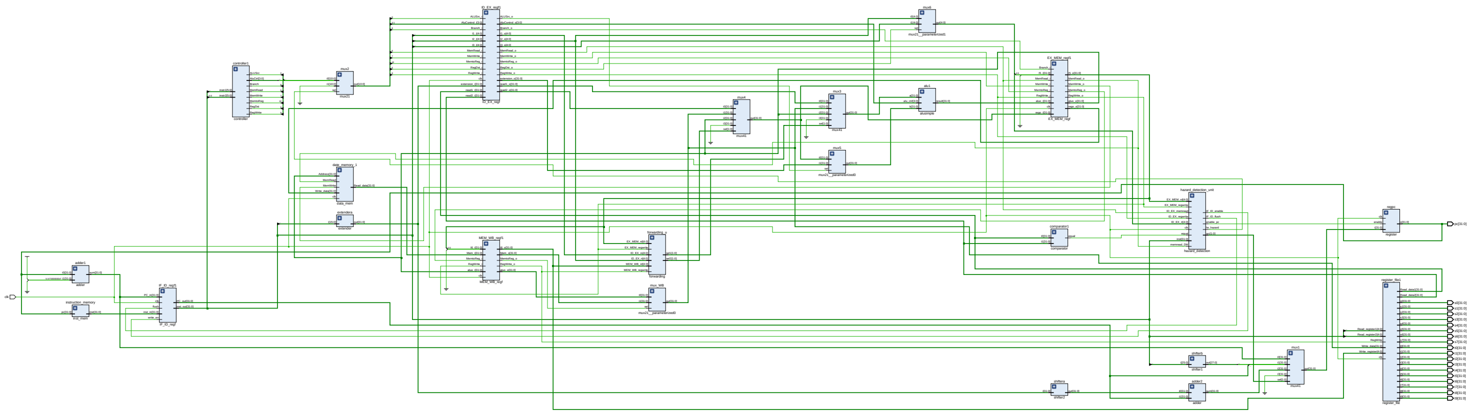
```
module ssd_driver_new(C, Q);
input [3:0] Q;
output [6:0] C;

wire CA,CB,CC,CD,CE,CF,CG;

ssd_driver (CA,CB,CC,CD,CE,CF,CG,Q);

assign C={CA,CB,CC,CD,CE,CF,CG};

endmodule
```

ssd_driver.v

```
module ssd_driver (CA,CB,CC,CD,CE,CF,CG,Q);
    input [3:0] Q;
    output CA,CB,CC,CD,CE,CF,CG;
    reg CA,CB,CC,CD,CE,CF,CG;

    always @(Q)
    begin
        if (Q==4'b0000) begin
            CA<=0;
            CB<=0;
            CC<=0;
            CD<=0;
            CE<=0;
            CF<=0;
            CG<=1;
        end
        else if (Q==4'b0001) begin
            CA<=1;
            CB<=0;
            CC<=0;
            CD<=1;
            CE<=1;
            CF<=1;
            CG<=1;
        end
        else if (Q==4'b0010) begin
            CA<=0;
            CB<=0;
            CC<=1;
            CD<=0;
            CE<=0;
            CF<=1;
            CG<=0;
        end
        else if (Q==4'b0011) begin
            CA<=0;
            CB<=0;
            CC<=0;
            CD<=0;
            CE<=1;
            CF<=1;
            CG<=0;
        end
        else if (Q==4'b0100) begin
            CA<=1;
            CB<=0;
            CC<=0;
            CD<=1;
            CE<=1;
            CF<=0;
            CG<=0;
        end
        else if (Q==4'b0101) begin
            CA<=0;
            CB<=1;
            CC<=0;
            CD<=0;
            CE<=1;
            CF<=0;
            CG<=0;
        end
        else if (Q==4'b0110) begin
            CA<=0;
            CB<=1;
            CC<=0;
            CD<=0;
            CE<=0;
            CF<=0;
            CG<=0;
        end
        else if (Q==4'b0111) begin
            CA<=0;
            CB<=0;
            CC<=0;
            CD<=1;
            CE<=1;
            CF<=1;
            CG<=1;
        end
        else if (Q==4'b1000) begin
            CA<=0;
            CB<=0;
            CC<=0;
            CD<=0;
            CE<=0;
            CF<=0;
            CG<=0;
        end
        else if (Q==4'b1001) begin
            CA<=0;
            CB<=0;
            CC<=0;
            CD<=0;
            CE<=1;
            CF<=0;
            CG<=0;
        end
        else if (Q==4'b1010) begin
            CA<=0;
            CB<=0;
            CC<=0;
            CD<=1;
            CE<=0;
            CF<=0;
            CG<=0;
        end
        else if (Q==4'b1011) begin
            CA<=1;
            CB<=1;
            CC<=0;
            CD<=0;
            CE<=0;
            CF<=0;
            CG<=0;
        end
        else if (Q==4'b1100) begin
            CA<=0;
            CB<=1;
            CC<=1;
            CD<=0;
            CE<=0;
            CF<=0;
            CG<=1;
        end
        else if (Q==4'b1101) begin
            CA<=1;
            CB<=0;
            CC<=0;
            CD<=0;
            CE<=0;
            CF<=1;
            CG<=0;
        end
        else if (Q==4'b1110) begin
            CA<=0;
            CB<=1;
            CC<=1;
            CD<=0;
            CE<=0;
            CF<=0;
            CG<=0;
        end
        else if (Q==4'b1111) begin
            CA<=0;
            CB<=1;
            CC<=1;
            CD<=1;
            CE<=0;
            CF<=0;
            CG<=0;
        end
    end



endmodule
```

Appendix B: RTL schematic of pipeline CPU

(On the next page)



RTL schematic for pipelined processor.

RTL schematic for FPGA implementation of the pipelined processor.