Introduction to Computer Organization VE370 *
Project II

Group 31
Jinyuan Chen 518370910100
Yuchuan Tian 518370910040
Qinzhe Yang 518370910006
Hanyun Zhao 518370910091

November 12, 2020

*UM-SJTU Joint Institute

Contents

1 Objectives

This project requires us to implement the pipelined processor in Verilog and demonstrate on FGPA board. Instructions needed to satisfy include:

1. *lw*, *sw*
2. *add*, *addi*, *sub*, *and*, *andi*, *or*, *slt*
3. *beq*, *bne*, *j*

Figure 1 shows the top-level block diagram of the pipelined implementation.

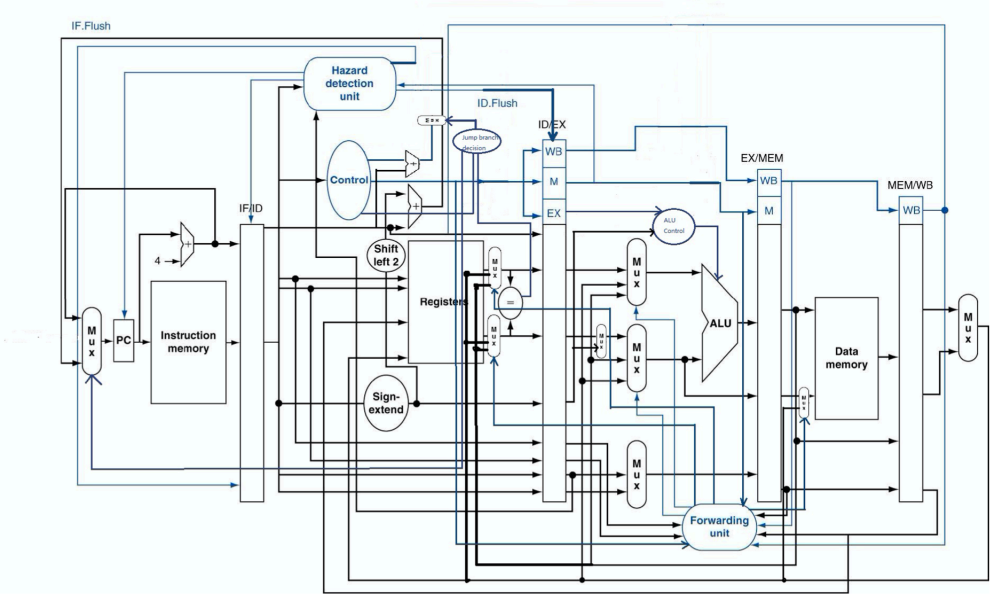


Figure 1: Pipelined Implementation of MIPS Architecture

2 Components

Note: All module Verilog codes can be found in Appendix A.

2.1 Instruction Memory

In this module, we first read the instructions from the text file (in board implementation, we can only initialize all instructions first by assigning each memory block with a certain instruction) and store them in IM. Then we output the target instruction code located at the input address in IM.

2.2 Registers

In this module, we read and write data in a 32-bit register file of size 32. Based on the clock signal and RegWrite signal, we read or write data in the registers with input target address. Writing is done in the first half of the clock cycle while reading is always available.

2.3 ALU

In this module, we conduct arithmetic operations of the inputs and output the result. Based on the ALUControl signal, we determine which specific operation to be executed.

2.4 Data Memory

In this module, we read or write data in the memory storage. Based on the MemWrite and MemRead signals, we determine whether to read or write the data in the input address of the storage. Both reading and writing are synchronous.

2.5 Forward Unit

This module has three outputs controlling the four inputs of the ALU block and comparator. ForwardA and ForwardB controls the input of ALU. ForwardC and ForwardD controls the input of comparator. Forward unit totally considers three hazards: EX hazard, MEM hazard and data hazard meeting with branch instructions.

2.6 Hazard Detection Unit

This module includes four outputs: PCWrite controlling whether PC is needed in write in, stall controlling whether pipeline needs a nop, IF_ID_Write controlling whether IF_ID registers needs to write in and IF_Flush controlling whether IF register needs to be cleaned up. Totally, hazard detection unit considers lw-use hazard and branch hazard. In addition, we need to take one special case into consideration that lw-use hazard meets with branch hazard.

2.7 IF/ID Register

In this register, we control the delivery of PC address and instruction code. Based on the clock signal, if the register receives ID_Flush with 1, a flush is applied and we set all outputs to 0. Otherwise, id the register receives IF_ID_Write with 1, we should write the output and deliver the content from IF stage to ID stage.

2.8 ID/EX Register

In this register, we deliver the data from ID stage to EX stage. Based on the clock signal, we write the output and deliver the content from ID stage to EX stage.

2.9 EX/MEM Register

In this register, we control the delivery of data from EX stage to MEM stage based on the clock signal. If the register receives EX_Flush with 1, a flush is applied and we set all outputs to 0. Otherwise, we write the output and deliver the content from EX stage to MEM stage.

2.10 MEM/WB Register

In this register, we deliver the data from MEM stage to WB stage. Based on the clock signal, we write the output and deliver the content from MEM stage to WB stage.

2.11 SSD Driver

In this register, we input 4-bit binary numbers and provide output signals to the seven anodes and cathodes of an SSD to display the corresponding decimal number in LED.

3 Assembly

We assembled all those components, according to the schematic in Figure 1 in software Xlinx Vivado. Then we generated the RTL Schematic of our pipeline processor, shown in Figure 2.

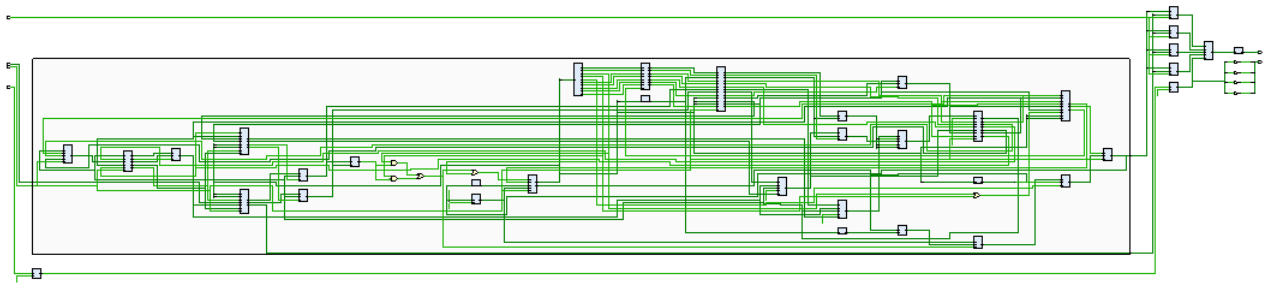


Figure 2: RTL Schematic

4 Simulation

4.1 Demo Code MIPS Instructions

```
1 00100000 00001000 00000000 00100000 //addi $t0, $zero, 0x20
2 00100000 00001001 00000000 00110111 //addi $t1, $zero, 0x37
3 00000001 00001001 10000000 00100100 //and $s0, $t0, $t1
4 00000001 00001001 10000000 00100101 //or $s0, $t0, $t1
5 10101100 00010000 00000000 00000100 //sw $s0, 4($zero)
6 10101100 00001000 00000000 00001000 //sw $t0, 8($zero)
```

```
00000001 00001001 10001000 00100000 //add $s1, $t0, $t1
00000001 00001001 10010000 00100010 //sub $s2, $t0, $t1
00010010 00110010 00000000 00001001 //beq $s1, $s2, error0
10001100 00010001 00000000 00000100 //lw $s1, 4($zero)
00110010 00110010 00000000 01001000 //andi $s2, $s1, 0x48
00010010 00110010 00000000 00001001 //beq $s1, $s2, error1
10001100 00010011 00000000 00001000 //lw $s3, 8($zero)
00010010 00010011 00000000 00001010 //beq $s0, $s3, error2
00000010 01010001 10100000 00101010 //slt $s4, $s2, $s1 (Last)
00010010 10000000 00000000 00001111 //beq $s4, $0, EXIT
00000010 00100000 10010000 00100000 //add $s2, $s1, $0
00001000 00000000 00000000 00001110 //j Last
00100000 00001000 00000000 00000000 //addi $t0, $0, 0(error0)
00100000 00001001 00000000 00000000 //addi $t1, $0, 0
00001000 00000000 00000000 00011111 //j EXIT
00100000 00001000 00000000 00000001 //addi $t0, $0, 1(error1)
00100000 00001001 00000000 00000001 //addi $t1, $0, 1
00001000 00000000 00000000 00011111 //j EXIT
00100000 00001000 00000000 00000010 //addi $t0, $0, 2(error2)
00100000 00001001 00000000 00000010 //addi $t1, $0, 2
00001000 00000000 00000000 00011111 //j EXIT
00100000 00001000 00000000 00000011 //addi $t0, $0, 3(error3)
00100000 00001001 00000000 00000011 //addi $t1, $0, 3
00001000 00000000 00000000 00011111 //j EXIT
```

4.2 Demo Code Results

Note: For the elegance and clarity of the result, in this part we used textual result directly copied from the Vivado Console. The screenshot results that prove our work are attached in Appendix C.

Our program starts at 0ns, logic low, and each clock cycle takes 20ns. Every time an instruction is loaded, the instant time, the PC address, and contents of some registers are displayed. Together with the given sequence of instructions, we'll explain what is happening as well as verify the results.

What need mentioning is that, since PC update is triggered by rising edge, and we define a period starts from logic low, that means we'll actually see all the changes in the middle of a period. To be clear, our periods are 0-20ns, 20-40ns 40-60ns and the changes happens in 10ns, 30ns, 50ns and so on.

4.2.1 1st to 5th Cycle

When the clock cycle is in 1 to 4, there isn't any result, and not until the 5^{th} cycle does the first instruction finish computing its result. The first instruction assign register t0 with 0x20.

```
Time:                   80, CLK=0, PC=00000010
[$s0] = 00000000, [$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:                   90, CLK=1, 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
```

4.2.2 6th Cycle

When clock cycle is 6, the sixth instruction enters IF stage, and the result of the second instruction is written back to register file. So register t1 is written to 0x37.

```
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] = 00000000, [$t2] = 00000000, [$t3] = 00000000
[$t4] = 00000000, [$t5] = 00000000, [$t6] = 00000000
[$t7] = 00000000, [$t8] = 00000000, [$t9] = 00000000
Time:                  110, CLK=1, 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
```

4.2.3 7th Cycle

When clock cycle is 7, the third instruction finish writing the registers. s0 is written to 0x20 & 0x37 = 0x20.

```
Time:                  120, CLK=0, PC=00000018
[$s0] = 00000000, [$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:                  130, CLK=1, 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
```

4.2.4 8th Cycle

When clock cycle is 8, the fourth instruction finishes. s0 is rewritten to 0x29 | 0x37 = 0x37. And at this cycle, a *sw* instrcution is in MEM stage and s0 is stored.

```
Time:                  140, CLK=0, PC=0000001c
[$s0] = 00000020, [$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:                  150, CLK=1, 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
```

4.2.5 9th Cycle

When clock cycle is 9, the first *sw* arrives in WB stage, the second *sw* is in MEM stage and t0 is stored.

```
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=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
```

4.2.6 10th Cycle

When clock cycle is 10, the sixth instruction *sw* reaches WB stage. Still no register changed. However a *beq* is in ID stage and a *sub* is in Ex stage. Data hazard is detected, a stall will be inserted in next cycle.

```
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
[$s0] = 00000037, [$s1] = 00000000, [$s2] = 00000000
[$s3] = 00000000, [$s4] = 00000000, [$s5] = 00000000
[$s6] = 00000000, [$s7] = 00000000, [$t0] = 00000020
```

```
12 [$t1] = 00000037, [$t2] = 00000000, [$t3] = 00000000
13 [$t4] = 00000000, [$t5] = 00000000, [$t6] = 00000000
14 [$t7] = 00000000, [$t8] = 00000000, [$t9] = 00000000
```

4.2.7 11th Cycle

When clock cycle is 11, the *add* instrcution finishes. s0=t0+t1=0x57. *beq* stays in ID stage, and a nop is inserted in Ex stage.

```
 1 Time:                  200, CLK=0, PC=00000024
 2 [$s0] = 00000037, [$s1] = 00000000, [$s2] = 00000000
 3 [$s3] = 00000000, [$s4] = 00000000, [$s5] = 00000000
 4 [$s6] = 00000000, [$s7] = 00000000, [$t0] = 00000020
 5 [$t1] = 00000037, [$t2] = 00000000, [$t3] = 00000000
 6 [$t4] = 00000000, [$t5] = 00000000, [$t6] = 00000000
 7 [$t7] = 00000000, [$t8] = 00000000, [$t9] = 00000000
 8 Time:                  210, CLK=1, PC=00000024
 9 [$s0] = 00000037, [$s1] = 00000057, [$s2] = 00000000
10 [$s3] = 00000000, [$s4] = 00000000, [$s5] = 00000000
11 [$s6] = 00000000, [$s7] = 00000000, [$t0] = 00000020
12 [$t1] = 00000037, [$t2] = 00000000, [$t3] = 00000000
13 [$t4] = 00000000, [$t5] = 00000000, [$t6] = 00000000
14 [$t7] = 00000000, [$t8] = 00000000, [$t9] = 00000000
```

4.2.8 12th Cycle

When the clock cycle is 12, the *sub* instruction finishes, s2=t0-t1=0x20-0x37=0xffffffe9. The *beq* branch is in Ex and confirmed not to execute.

```
 1 Time:                  220, CLK=0, PC=00000028
 2 [$s0] = 00000037, [$s1] = 00000057, [$s2] = 00000000
 3 [$s3] = 00000000, [$s4] = 00000000, [$s5] = 00000000
 4 [$s6] = 00000000, [$s7] = 00000000, [$t0] = 00000020
 5 [$t1] = 00000037, [$t2] = 00000000, [$t3] = 00000000
 6 [$t4] = 00000000, [$t5] = 00000000, [$t6] = 00000000
 7 [$t7] = 00000000, [$t8] = 00000000, [$t9] = 00000000
 8 Time:                  230, CLK=1, PC=00000028
 9 [$s0] = 00000037, [$s1] = 00000057, [$s2] = ffffffe9
10 [$s3] = 00000000, [$s4] = 00000000, [$s5] = 00000000
11 [$s6] = 00000000, [$s7] = 00000000, [$t0] = 00000020
12 [$t1] = 00000037, [$t2] = 00000000, [$t3] = 00000000
13 [$t4] = 00000000, [$t5] = 00000000, [$t6] = 00000000
14 [$t7] = 00000000, [$t8] = 00000000, [$t9] = 00000000
```

4.2.9 13th Cycle

When the clock cycle is 13, the *nop* reaches WB stage. Meanwhile, the 10th instruction *lw* is in Ex stage, and the 11th instruction *andi* is in ID stage. *lw* use hazard is detected, so a nop will be inserted in the next cycle.

```
1 Time:                  240, CLK=0, PC=0000002c
2 [$s0] = 00000037, [$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, [$t9] = 00000000
Time:                   250, CLK=1, PC=0000002c
[$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
```

4.2.10 14th Cycle

When the clock cycle is 14, the *beq* instruction reaches WB stage. Since it does not branch, no register is changed.

```
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
[$s0] = 00000037, [$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
```

4.2.11 15th Cycle

When the clock cycle is 15, the *lw* instruction finishes. 0x37 is loaded into register s1. In addition, a *beq* instruction is in ID stage. Data dependency is detected and a *nop* will be inserted into Ex stage in the next cycle.

```
Time:                   280, 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:                   290, CLK=1, 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
```

4.2.12 16th Cycle

When clock cycle is 16, the *nop* instruction reaches WB stage. Therefore, no register is changed. Also, a *nop* is inserted in the EX stage.

```
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
[$s0] = 00000037, [$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
```

4.2.13 17th Cycle

When clock cycle is 17, the 14th instruction is in PC. The register is shown as following:

```
Time:                  320, CLK=0, PC=00000034
[$s0] = 00000037, [$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=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
```

The 11th instruction is currently in the WB stage. This instruction writes back register s2 as 0.

4.2.14 18th Cycle

When clock cycle is 18, the 15th instruction is in PC. The register is shown as following:

```
Time:                  340, 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:                  350, CLK=1, 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
```

The WB stage is currently a nop, therefore nothing will be writtern back.

4.2.15 19th and 20th Cycle

When clock cycle is 19 and 20, no new instructions will go through the pipeline, because the 13th and 14th instruction has data hazard. *lw* instuction and *beq* instruction had to add two nops between them. However the remaining instructions in the pipeline were still running. Hence, when clock cycle is 19, the 12th instruction is in WB stage and doing nothing. When clock cycle is 20, the register is shown following:

```
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
[$s0] = 00000037, [$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
```

The 13th instuction is in WB stage and write register s3 as 0x20.

4.2.16 21st Cycle

When clock cycle is 21, the 16th instruction is in PC. The register is shown as following:

```
Time:                  400, 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:                  410, CLK=1, 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
```

At this moment, WB stage is a nop. Hence nothing will be written back.

4.2.17 22nd Cycle

When clock cycle is 22, the 17th instruction is in PC. The register is shown as following:

```
Time:                  420, 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:                  430, CLK=1, 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
```

The WB stage is still a nop at this time.

4.2.18 23rd Cycle

When clock cycle is 23, no new instruction is going into PC. This is because the 15th and 16th instruction has data hazard. R-type instruction and branch instruction should have one nop between them. The register is shown following:

```
Time:                  440, 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:                  450, CLK=1, 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
```

At this time, the 14th instruction is in the WB stage and has nothing to write back.

4.2.19 24th Cycle

When clock cycle is 24, the 18th instruction is in PC. The register is shown as following:

```
Time:                  460, CLK=0, PC=00000044
[$s0] = 00000037, [$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=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
```

The 15th instruction is in the WB stage, writing register s4 as 0x1.

4.2.20 25th Cycle

When clock cycle is 25, the 19th instruction is in the PC, which is 0x00000048.

```
Time:                  480, 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:                  490, CLK=1, 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
```

At this moment, there is no register value change. The nop instruction is at WB stage and has nothing to write back.

4.2.21 26th Cycle

When clock cycle is 26, jump instruction at the 18th instruction is activated. Hence, the previous 19th instruction is flushed and nop reaches ID stage. At IF stage, the jump target 0x00000038 is loaded in PC at IF stage.

```
Time:                  500, 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:                  500, 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
```

At this moment, there is no register value change. The 16th instruction is at WB stage and has nothing to write back.

4.2.22 27th Cycle

When clock cycle is 27, the 16th instruction is in the PC, which is 0x0000003c.

```
Time:                   520, CLK=0, PC=0000003c
[$s0] = 00000037, [$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=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
```

At this moment, register s2 changes from value 0x0 to 0x37. The 17th instruction is in the WB stage, writing register s2 as 0x37.

4.2.23 28th Cycle

When clock cycle is 28, the 17th instruction is in the PC, which is 0x00000040.

```
Time:                   540, 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:                   550, CLK=1, 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
```

At this moment, there is no register value change. The 18th instruction is at WB stage and has nothing to write back.

4.2.24 29th Cycle

When clock cycle is 29, the 17th instruction remains in the PC, which is still 0x00000040.

```
Time:                   560, 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:                  570, CLK=1, 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
```

At this moment, there is no register value change. The nop instruction is at WB stage and has nothing to write back.

4.2.25 30th Cycle

When clock cycle is 30, the 16th instruction beq is activated, so EXIT address is loaded into PC, which is 0x0000007c.

```
Time:                  580, CLK=0, PC=0000007c
[$s0] = 00000037, [$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=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
```

At this moment, register s4 changes from 0x1 to 0x0. The 15th instruction is in the WB stage, writing register s4 as 0x0.

4.2.26 31st to 34th Cycle

From then on, address adds 4 for each cycle; since those PC addresses are undefined, there will be no new instructions loaded. By this time (cycle 31), the remaining instructions are: a flushed nop in EX stage, the 16th instruction in MEM stage and a stalled nop in WB stage. Non of those instructions actually write something in the register file. So the register file has the final result:

```
[$s0] = 00000037, [$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
```

4.3 BNE instruction

The BNE instruction is not included in the Demo Code, so we wrote a demo code that includes bne instruction, attached here:

```
00010100 00000000 00000000 00001111 // bne $zero $zero EXIT
00100000 00001000 00000000 00100000 // addi $t0, $zero, 0x20
00010101 00000000 00000000 00001111 // bne $t0 $zero EXIT
00100000 00001001 00000000 00100000 // addi $t1, $zero, 0x20
00100000 00001001 00000000 00100000 // addi $t1, $zero, 0x20
00100000 00001001 00000000 00100000 // addi $t1, $zero, 0x20
00100000 00001001 00000000 00100000 // addi $t1, $zero, 0x20
```

Correspondingly, we modified the initialization of our Instruction Memory module. Then the simulation result turns out to be the following.

4.3.1 1st to 3rd Cycle

```
Time:           0, CLK=           0, PC=00000000
[$s0] = 00000000, [$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=00000000
[$s0] = 00000000, [$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=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
```

At 2nd cycle, the 1st instruction (bne) is at the ID stage. Bne instruction should not be activated as $zero is equal to $zero, and hence, the next instruction becomes PC=00000008.

```
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, [$t9] = 00000000
Time:                   50, CLK=1, PC=00000008
[$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
```

4.3.2 4th to 6th Cycle

In the 4th cycle, the 3rd instruction (namely, the second bne) is at stage ID. Though the new value of $t0: 0x20 hasn't been stored yet, due to forwarding and hazard detection, one nop is released at 5th cycle. Hence PC remains unchanged for two cycles, waiting for the 2nd instruction to reach MEM stage.

```
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=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:                   80, 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:                   90, CLK=1, 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
```

Then at 5th cycle, when the value of $t1 is ready at MEM stage, it is propagated back to ID stage and then bne is activated. Hence in 6th cycle, PC jumps to EXIT. Note here at the 6th cycle that the value of $t0 is changed to 0x20 at posedge because here the 2nd instruction reaches WB stage.

```
Time:                  100, CLK=0, PC=00000048
[$s0] = 00000000, [$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:                  110, CLK=1, PC=00000048
[$s0] = 00000000, [$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
```

4.3.3 After 6th Cycle

```
Time:                  120, CLK=0, PC=0000004c
[$s0] = 00000000, [$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:                  130, CLK=1, PC=0000004c
[$s0] = 00000000, [$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:                  140, CLK=0, PC=00000050
[$s0] = 00000000, [$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:                  150, CLK=1, PC=00000050
[$s0] = 00000000, [$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
```

We can see that $t1 is never changed; hence the addi instruction after the 2nd bne is successfully flushed. Those results prove that bne instructions are effective in both activated and inactivated cases.

5 Conclusions

In this project, we finish the pipelined implementation of MIPS computer in Verilog and demonstrate it on FPGA board. It supports a subset of MIPS instruction set including the memory-reference instructions load word (lw) and store word (sw); the arithmetic-logical in-

structions add, addi, sub, and, andi, or, and slt; and the jumping instructions branch equal (beq), branch not equal (bne), and jump (j).

A pipelined implementation saves time and raises efficiency compared with the single cycle implementation, yet it requires special attention to solving data hazard problems. Our implementation method and simulation results are shown and explained in the previous parts. And the results satisfy what we expected.

A Source Codes

A.1 PC Register

```
1  module PC(PCout,PCin,PCWrite,clk);
2
3  output [31:0] PCout;
4  input [31:0] PCin;
5  input PCWrite;
6  input clk;
7  reg [31:0] PCout=32'b0;
8
9  always @ (posedge clk) begin
10     if (PCWrite == 1'b1) begin
11         PCout=PCin;
12     end
13 end
14
15 endmodule
```

A.2 Adder

```
1  module Adder(result,a,b);
2
3  output [31:0] result;
4  input [31:0] a;
5  input [31:0] b;
6  reg [1:0] operation=2'b10;
7  reg ainvert=1'b0;
8  reg bnegate=1'b0;
9  wire zero;
10 wire overflow;
11
12 ALU tempALU (result,overflow,zero,a,b,ainvert,bnegate,operation);
13
14 endmodule
```

A.3 Instruction Memory

```
1  module instructionMem(Instruction,Readaddr);
2
3  output [31:0] Instruction;
4  input [31:0] Readaddr;
5  wire [31:0] Instruction;
6  reg [7:0] memory [63:0][3:0];
7  integer i;
8
9  //BONUS:
10 initial begin
11     { memory[0][3], memory[0][2], memory[0][1], memory[0][0] } = 32'
           b00100000000010000000000000100000; // 1
12     { memory[1][3], memory[1][2], memory[1][1], memory[1][0] } = 32'
           b00100000000010010000000000110111; // 2
```

```
13      { memory[2][3], memory[2][2], memory[2][1], memory[2][0] } = 32'
            b00000001000010011000000000100100; // 3
14      { memory[3][3], memory[3][2], memory[3][1], memory[3][0] } = 32'
            b00000001000010011000000000100101; // 4
15      { memory[4][3], memory[4][2], memory[4][1], memory[4][0] } = 32'
            b10101100000100000000000000000100; // 5
16      { memory[5][3], memory[5][2], memory[5][1], memory[5][0] } = 32'
            b10101100000010000000000000001000; // 6
17      { memory[6][3], memory[6][2], memory[6][1], memory[6][0] } = 32'
            b00000001000010011000100000100000; // 7
18      { memory[7][3], memory[7][2], memory[7][1], memory[7][0] } = 32'
            b00000001000010011001000000100010; // 8
19      { memory[8][3], memory[8][2], memory[8][1], memory[8][0] } = 32'
            b00010010001100100000000000001001; // 9
20      { memory[9][3], memory[9][2], memory[9][1], memory[9][0] } = 32'
            b10001100000100010000000000000100; // 10
21      { memory[10][3], memory[10][2], memory[10][1], memory[10][0] } = 32'
            b00110010001100100000000001001000; // 11
22      { memory[11][3], memory[11][2], memory[11][1], memory[11][0] } = 32'
            b00010010001100100000000000001001; // 12
23      { memory[12][3], memory[12][2], memory[12][1], memory[12][0] } = 32'
            b10001100000100110000000000001000; // 13
24      { memory[13][3], memory[13][2], memory[13][1], memory[13][0] } = 32'
            b00010010000100110000000000001010; // 14
25      { memory[14][3], memory[14][2], memory[14][1], memory[14][0] } = 32'
            b00000010010100011010000000101010; // 15
26      { memory[15][3], memory[15][2], memory[15][1], memory[15][0] } = 32'
            b00010010100000000000000000001111; // 16
27      { memory[16][3], memory[16][2], memory[16][1], memory[16][0] } = 32'
            b00000010001000001001000000100000; // 17
28      { memory[17][3], memory[17][2], memory[17][1], memory[17][0] } = 32'
            b00001000000000000000000000001110; // 18
29      { memory[18][3], memory[18][2], memory[18][1], memory[18][0] } = 32'
            b00100000000010000000000000000000; // 19
30      { memory[19][3], memory[19][2], memory[19][1], memory[19][0] } = 32'
            b00100000000010010000000000000000; // 20
31      { memory[20][3], memory[20][2], memory[20][1], memory[20][0] } = 32'
            b00001000000000000000000000011111; // 21
32      { memory[21][3], memory[21][2], memory[21][1], memory[21][0] } = 32'
            b00100000000010000000000000000001; // 22
33      { memory[22][3], memory[22][2], memory[22][1], memory[22][0] } = 32'
            b00100000000010010000000000000001; // 23
34      { memory[23][3], memory[23][2], memory[23][1], memory[23][0] } = 32'
            b00001000000000000000000000011111; // 24
35      { memory[24][3], memory[24][2], memory[24][1], memory[24][0] } = 32'
            b00100000000010000000000000000010; // 25
36      { memory[25][3], memory[25][2], memory[25][1], memory[25][0] } = 32'
            b00100000000010010000000000000010; // 26
37      { memory[26][3], memory[26][2], memory[26][1], memory[26][0] } = 32'
            b00001000000000000000000000011111; // 27
38      { memory[27][3], memory[27][2], memory[27][1], memory[27][0] } = 32'
            b00100000000010000000000000000011; // 28
39      { memory[28][3], memory[28][2], memory[28][1], memory[28][0] } = 32'
            b00100000000010010000000000000011; // 29
```

```
40      { memory[29][3], memory[29][2], memory[29][1], memory[29][0] } = 32'
            b00001000000000000000000000011111; // 30
41
42      for ( i = 30; i < 64; i = i+1 ) begin
43          memory[i][3] = 8'b0;
44          memory[i][2] = 8'b0;
45          memory[i][1] = 8'b0;
46          memory[i][0] = 8'b0;
47      end
48  end
49
50  // direction need to be inversed
51  assign Instruction[31:24]=memory[Readaddr/4][3];
52  assign Instruction[23:16]=memory[Readaddr/4][2];
53  assign Instruction[15:8]=memory[Readaddr/4][1];
54  assign Instruction[7:0]=memory[Readaddr/4][0];
55
56  endmodule
```

A.4 IF/ID Register

```
1   module IFID(clk, IF_ID_Write, ID_Flush, IF_PCplusFour, IF_Instruction,
        ID_PCplusFour, ID_Instruction);
2
3   input clk;
4   input IF_ID_Write;
5   input ID_Flush;
6   input [31:0] IF_PCplusFour;
7   input [31:0] IF_Instruction;
8   output [31:0] ID_PCplusFour;
9   output [31:0] ID_Instruction;
10  reg [31:0] ID_PCplusFour=32'b0;
11  reg [31:0] ID_Instruction=32'b0;
12
13  always @(posedge clk) begin
14      if (ID_Flush == 1'b1) begin
15          ID_PCplusFour <= 32'b0;
16          ID_Instruction <= 32'b0;
17      end
18      else if (IF_ID_Write == 1'b1) begin
19          ID_PCplusFour <= IF_PCplusFour;
20          ID_Instruction <= IF_Instruction;
21      end
22  end
23
24  endmodule
```

A.5 Hazard Detection Unit

```
1   module Hazard_Detection_Unit(IF_ID_RegisterRs, IF_ID_RegisterRt,
        ID_EX_RegisterRt, ID_EX_RegisterRd, EX_MEM_RegisterRd, ID_EX_MemRead,
        EX_MEM_MemRead, ID_EX_RegWrite, IDbranch, PCWrite, IF_ID_Write, stall,
        IF_Flush);
2
```

```
3  input [4:0] IF_ID_RegisterRs;
4  input [4:0] IF_ID_RegisterRt;
5  input [4:0] ID_EX_RegisterRt;
6  input [4:0] ID_EX_RegisterRd;
7  input [4:0] EX_MEM_RegisterRd;
8  input ID_EX_MemRead;
9  input EX_MEM_MemRead;
10 input ID_EX_RegWrite;
11 input IDbranch;
12 output PCWrite;
13 output IF_ID_Write;
14 output stall;
15 output IF_Flush;
16 reg PCWrite=1'b1;
17 reg IF_ID_Write=1'b1;
18 reg stall=1'b0;
19 reg IF_Flush=1'b0;
20
21 initial begin
22     PCWrite = 1'b1;
23     IF_ID_Write = 1'b1;
24     stall = 1'b0;
25 end
26
27 always @(*) begin
28     //PC write and IFID write
29     if (ID_EX_MemRead && ((ID_EX_RegisterRt == IF_ID_RegisterRs) || (
           ID_EX_RegisterRt == IF_ID_RegisterRt))) begin
30         PCWrite = 1'b0;
31         IF_ID_Write = 1'b0;
32     end
33     else begin
34         PCWrite = 1'b1;
35         IF_ID_Write = 1'b1;
36     end
37     //IF_Flush
38     if (IDbranch) begin
39         if ((ID_EX_RegWrite && ID_EX_RegisterRd && ((ID_EX_RegisterRd ==
               IF_ID_RegisterRs) || (ID_EX_RegisterRd == IF_ID_RegisterRt)))
               || (EX_MEM_MemRead && EX_MEM_RegisterRd && ((EX_MEM_RegisterRd
               == IF_ID_RegisterRs) || (EX_MEM_RegisterRd == IF_ID_RegisterRt)
               )) ) begin
40             stall=1'b1;
41             PCWrite = 1'b0;
42             IF_ID_Write = 1'b0;
43         end
44         else begin
45             stall=1'b0;
46             PCWrite = 1'b1;
47             IF_ID_Write = 1'b1;
48         end
49     end
50
51     //stall
```

```
52      if  ( (ID_EX_MemRead && ((ID_EX_RegisterRt == IF_ID_RegisterRs) || (
            ID_EX_RegisterRt == IF_ID_RegisterRt))) || (IDbranch) && ((
            ID_EX_RegWrite && ID_EX_RegisterRd && ((ID_EX_RegisterRd ==
            IF_ID_RegisterRs) || (ID_EX_RegisterRd == IF_ID_RegisterRt))) || (
            EX_MEM_MemRead && EX_MEM_RegisterRd && ((EX_MEM_RegisterRd ==
            IF_ID_RegisterRs) || (EX_MEM_RegisterRd == IF_ID_RegisterRt))) ) )
            begin
53          stall=1'b1;
54          PCWrite = 1'b0;
55          IF_ID_Write = 1'b0;
56      end
57      else begin
58          stall=1'b0;
59          PCWrite = 1'b1;
60          IF_ID_Write = 1'b1;
61      end
62  end
63
64  endmodule
```

A.6 Control

```
1   module Control(ALUOp,RegDst,Jump,Branch,MemRead,MemtoReg,MemWrite,ALUSrc,
        RegWrite,BranchNot,Opcode);
2
3   output RegDst,Jump,Branch,MemRead,MemtoReg,MemWrite,ALUSrc,RegWrite,
        BranchNot;
4   output [1:0] ALUOp;
5   input [5:0] Opcode;
6   reg RegDst=1'b0;
7   reg Jump=1'b0;
8   reg Branch=1'b0;
9   reg MemRead=1'b0;
10  reg MemtoReg=1'b0;
11  reg MemWrite=1'b0;
12  reg ALUSrc=1'b0;
13  reg RegWrite=1'b0;
14  reg BranchNot=1'b0;
15  reg [1:0] ALUOp=2'b0;
16
17  always @ (Opcode) begin
18      case(Opcode)
19          6'b000000:begin
20              ALUOp=2'b10;RegDst=1'b1;Jump=1'b0;Branch=1'b0;MemRead=1'b0;
                    MemtoReg=1'b0;MemWrite=1'b0;ALUSrc=1'b0;RegWrite=1'b1;
                    BranchNot=1'b0;
21          end
22          6'b100011:begin
23              ALUOp=2'b00;RegDst=1'b0;Jump=1'b0;Branch=1'b0;MemRead=1'b1;
                    MemtoReg=1'b1;MemWrite=1'b0;ALUSrc=1'b1;RegWrite=1'b1;
                    BranchNot=1'b0;
24          end
25          6'b101011:begin
26              ALUOp=2'b00;RegDst=1'b0;Jump=1'b0;Branch=1'b0;MemRead=1'b0;
```

```
                MemtoReg=1'b0;MemWrite=1'b1;ALUSrc=1'b1;RegWrite=1'b0;
                BranchNot=1'b0;
        end
        6'b001000:begin
            ALUOp=2'b00;RegDst=1'b0;Jump=1'b0;Branch=1'b0;MemRead=1'b0;
                MemtoReg=1'b0;MemWrite=1'b0;ALUSrc=1'b1;RegWrite=1'b1;
                BranchNot=1'b0;
        end
        6'b001100:begin
            ALUOp=2'b11;RegDst=1'b0;Jump=1'b0;Branch=1'b0;MemRead=1'b0;
                MemtoReg=1'b0;MemWrite=1'b0;ALUSrc=1'b1;RegWrite=1'b1;
                BranchNot=1'b0;
        end
        6'b000100:begin
            ALUOp=2'b01;RegDst=1'b0;Jump=1'b0;Branch=1'b1;MemRead=1'b0;
                MemtoReg=1'b0;MemWrite=1'b0;ALUSrc=1'b0;RegWrite=1'b0;
                BranchNot=1'b0;
        end
        6'b000101:begin
            ALUOp=2'b01;RegDst=1'b0;Jump=1'b0;Branch=1'b0;MemRead=1'b0;
                MemtoReg=1'b0;MemWrite=1'b0;ALUSrc=1'b0;RegWrite=1'b0;
                BranchNot=1'b1;
        end
        6'b000010:begin
            ALUOp=2'b00;RegDst=1'b0;Jump=1'b1;Branch=1'b0;MemRead=1'b0;
                MemtoReg=1'b0;MemWrite=1'b0;ALUSrc=1'b0;RegWrite=1'b0;
                BranchNot=1'b0;
        end
    endcase
end

endmodule
```

A.7 Mux Control

```
module muxControl(outALUOp,outRegDst,outALUSrc,outMemRead,outMemWrite,
    outRegWrite,outMemtoReg,inALUOp,inRegDst,inALUSrc,inMemRead,inMemWrite,
    inRegWrite,inMemtoReg,s);

output [1:0] outALUOp;//EX
output outRegDst,outALUSrc;
output outMemRead,outMemWrite;
output outRegWrite,outMemtoReg;
input [1:0] inALUOp;//EX
input inRegDst,inALUSrc;
input inMemRead,inMemWrite;
input inRegWrite,inMemtoReg;
input s;
reg [1:0] outALUOp=2'b0;//EX
reg outRegDst=1'b0;
reg outALUSrc=1'b0;
reg outMemRead=1'b0;//MEM
reg outMemWrite=1'b0;
reg outRegWrite=1'b0;//WB
```

```
18 reg outMemtoReg=1'b0;
19
20 always @ (inALUOp or inRegDst or inALUSrc or inMemRead or inMemWrite or
       inRegWrite or inMemtoReg or s) begin
21     case(s)
22         1'b0: begin
23             outALUOp=inALUOp;
24             outRegDst=inRegDst;
25             outALUSrc=inALUSrc;
26             outMemRead=inMemRead;
27             outMemWrite=inMemWrite;
28             outRegWrite=inRegWrite;
29             outMemtoReg=inMemtoReg;
30         end
31         1'b1: begin
32             outALUOp=2'b00;//EX
33             outRegDst=1'b0;
34             outALUSrc=1'b0;
35             outMemRead=1'b0;
36                     outMemWrite=1'b0;
37                     outRegWrite=1'b0;
38     outMemtoReg=1'b0;
39     end
40     endcase
41 end
42 endmodule
```

A.8 Shift Left 16

```
1 module shiftLeft16(extendedImm,Imm);
2
3 input [31:0] Imm;
4 output [31:0] extendedImm;
5 reg [31:0] extendedImm=32'b0;
6
7 always @ (Imm) begin
8     extendedImm[31:2]=Imm[29:0];
9     extendedImm[1]=1'b0;
10    extendedImm[0]=1'b0;
11 end
12
13 endmodule
```

A.9 Shift Left 26

```
1 module shiftLeft26(extendedImm,Imm);
2
3 input [25:0] Imm;
4 output [27:0] extendedImm;
5 reg [27:0] extendedImm;
6
7 always @ (Imm) begin
8     extendedImm[27:2]=Imm;
9     extendedImm[1]=1'b0;
```

```
10      extendedImm[0]=1'b0;
11  end
12
13  endmodule
```

A.10 Sign Extend

```
1   module signExtend(extendedImm,Imm);
2
3   input [15:0] Imm;
4   output [31:0] extendedImm;
5   reg [31:0] extendedImm=32'b0;
6
7   always @ (Imm) begin
8       extendedImm[15:0]=Imm;
9       extendedImm[31]=Imm[15];
10      extendedImm[30]=Imm[15];
11      extendedImm[29]=Imm[15];
12      extendedImm[28]=Imm[15];
13      extendedImm[27]=Imm[15];
14      extendedImm[26]=Imm[15];
15      extendedImm[25]=Imm[15];
16      extendedImm[24]=Imm[15];
17      extendedImm[23]=Imm[15];
18      extendedImm[22]=Imm[15];
19      extendedImm[21]=Imm[15];
20      extendedImm[20]=Imm[15];
21      extendedImm[19]=Imm[15];
22      extendedImm[18]=Imm[15];
23      extendedImm[17]=Imm[15];
24      extendedImm[16]=Imm[15];
25  end
26
27  endmodule
```

A.11 Register File

```
1   module registers(Readdata1,Readdata2,Readreg1,Readreg2,Writereg,Writedata,
        RegWrite,clk,VisData,Visreg);
2
3   output [31:0] Readdata1;
4   output [31:0] Readdata2;
5   input RegWrite,clk;
6   input [4:0] Readreg1;
7   input [4:0] Readreg2;
8   input [4:0] Visreg;
9   output [31:0] VisData;
10  input [4:0] Writereg;
11  input [31:0] Writedata;
12  reg [31:0] storage [0:31];
13  reg [31:0] Readdata1=32'b0;
14  reg [31:0] Readdata2=32'b0;
15  reg [31:0] VisData;
16
```

```
17 //initialize storage
18 initial begin
19     storage[0]=32'b0;
20     storage[1]=32'b0;
21     storage[2]=32'b0;
22     storage[3]=32'b0;
23     storage[4]=32'b0;
24     storage[5]=32'b0;
25     storage[6]=32'b0;
26     storage[7]=32'b0;
27     storage[8]=32'b0;
28     storage[9]=32'b0;
29     storage[10]=32'b0;
30     storage[11]=32'b0;
31     storage[12]=32'b0;
32     storage[13]=32'b0;
33     storage[14]=32'b0;
34     storage[15]=32'b0;
35     storage[16]=32'b0;
36     storage[17]=32'b0;
37     storage[18]=32'b0;
38     storage[19]=32'b0;
39     storage[20]=32'b0;
40     storage[21]=32'b0;
41     storage[22]=32'b0;
42     storage[23]=32'b0;
43     storage[24]=32'b0;
44     storage[25]=32'b0;
45     storage[26]=32'b0;
46     storage[27]=32'b0;
47     storage[28]=32'b0;
48     storage[29]=32'b0;
49     storage[30]=32'b0;
50     storage[31]=32'b0;
51 end
52
53 // Notes: changed to "write first and then read"!!!
54 always @ (*) begin
55     Readdata1=storage[Readreg1];
56     Readdata2=storage[Readreg2];
57     VisData=storage[Visreg];
58 end
59
60 always @ (posedge clk or negedge clk) begin
61     $display("[$s0] = %h, [$s1] = %h, [$s2] = %h", storage[16], storage
           [17], storage[18]);
62     $display("[$s3] = %h, [$s4] = %h, [$s5] = %h", storage[19], storage
           [20], storage[21]);
63     $display("[$s6] = %h, [$s7] = %h, [$t0] = %h", storage[22], storage
           [23], storage[8]);
64     $display("[$t1] = %h, [$t2] = %h, [$t3] = %h", storage[9], storage[10],
            storage[11]);
65     $display("[$t4] = %h, [$t5] = %h, [$t6] = %h", storage[12], storage
           [13], storage[14]);
```

```
66      $display("[$t7] = %h, [$t8] = %h, [$t9] = %h", storage[15], storage
            [24], storage[25]);
67  end
68
69  always @ (negedge clk) begin
70      if (RegWrite==1) storage[Writereg]<=Writedata;
71  end
72
73  endmodule
```

A.12 Equal

```
1   module Equal(EqualRes,ReadData1, ReadData2);
2
3   output EqualRes;
4   input [31:0] ReadData1;
5   input [31:0] ReadData2;
6   reg EqualRes=1'b0;
7
8   always @(*) begin
9       if (ReadData1 == ReadData2) begin
10          EqualRes = 1'b1;
11      end
12      else begin
13          EqualRes = 1'b0;
14      end
15  end
16
17  endmodule
```

A.13 ID/EX Register

```
1   module IDEX(clk, ID_MemRead, ID_RegDst, ID_MemWrite, ID_MemtoReg,
        ID_RegWrite, ID_ReadData1, ID_ReadData2, ID_ExtendedImm, ID_RegRs,
        ID_RegRt, ID_RegRd, ID_ALUOp, ID_ALUSrc, EX_MemRead,EX_RegDst,
        EX_MemWrite, EX_MemtoReg, EX_RegWrite, EX_ReadData1, EX_ReadData2,
        EX_ExtendedImm, EX_RegRs, EX_RegRt, EX_RegRd, EX_ALUOp, EX_ALUSrc);
2
3   input clk;
4   input ID_MemRead;
5   input ID_RegDst;
6   input ID_MemWrite;
7   input ID_MemtoReg;
8   input ID_RegWrite;
9   input ID_ALUSrc;
10  input [31:0] ID_ReadData1;
11  input [31:0] ID_ReadData2;
12  input [31:0] ID_ExtendedImm;
13  input [4:0] ID_RegRs;
14  input [4:0] ID_RegRt;
15  input [4:0] ID_RegRd;
16  input [1:0] ID_ALUOp;
17  output EX_MemRead;
18  output EX_RegDst;
```

```
19 output EX_MemWrite;
20 output EX_MemtoReg;
21 output EX_RegWrite;
22 output EX_ALUSrc;
23 output [31:0] EX_ReadData1;
24 output [31:0] EX_ReadData2;
25 output [31:0] EX_ExtendedImm;
26 output [4:0] EX_RegRs;
27 output [4:0] EX_RegRt;
28 output [4:0] EX_RegRd;
29 output [1:0] EX_ALUOp;
30 reg EX_MemRead=1'b0;
31 reg EX_RegDst=1'b0;
32 reg EX_MemWrite=1'b0;
33 reg EX_MemtoReg=1'b0;
34 reg EX_RegWrite=1'b0;
35 reg EX_ALUSrc=1'b0;
36 reg [31:0] EX_ReadData1=32'b0;
37 reg [31:0] EX_ReadData2=32'b0;
38 reg [31:0] EX_ExtendedImm=32'b0;
39 reg [4:0] EX_RegRs=5'b00000;
40 reg [4:0] EX_RegRt=5'b00000;
41 reg [4:0] EX_RegRd=5'b00000;
42 reg [1:0] EX_ALUOp=2'b00;
43
44 always @(posedge clk) begin
45     EX_MemRead = ID_MemRead;
46     EX_RegDst = ID_RegDst;
47     EX_MemWrite = ID_MemWrite;
48     EX_MemtoReg = ID_MemtoReg;
49     EX_RegWrite = ID_RegWrite;
50     EX_ALUSrc = ID_ALUSrc;
51     EX_ReadData1 = ID_ReadData1;
52     EX_ReadData2 = ID_ReadData2;
53     EX_ExtendedImm = ID_ExtendedImm;
54     EX_RegRs = ID_RegRs;
55     EX_RegRt = ID_RegRt;
56     EX_RegRd = ID_RegRd;
57     EX_ALUOp = ID_ALUOp;
58 end
59
60 endmodule
```

A.14 ALU

```
1 module ALU(result,overflow,zero,a,b,ainvert,bnegate,operation);
2
3 input [31:0] a;
4 input [31:0] b;
5 input ainvert,bnegate;
6 input [1:0] operation;
7 output overflow,zero;
8 output [31:0] result;
9 wire [31:0] carryconnection;
```

```
10 wire [31:0] setconnection;
11 wire [31:0] overflowconnection;
12 wire invzero;
13 reg constLess=1'b0;
14
15 onebitALU A0 (result[0],carryconnection[0],setconnection[0],
       overflowconnection[0],a[0],b[0],ainvert,bnegate,setconnection[31],
       bnegate,operation);
16 onebitALU A1 (result[1],carryconnection[1],setconnection[1],
       overflowconnection[1],a[1],b[1],ainvert,bnegate,constLess,
       carryconnection[0],operation);
17 onebitALU A2 (result[2],carryconnection[2],setconnection[2],
       overflowconnection[2],a[2],b[2],ainvert,bnegate,constLess,
       carryconnection[1],operation);
18 onebitALU A3 (result[3],carryconnection[3],setconnection[3],
       overflowconnection[3],a[3],b[3],ainvert,bnegate,constLess,
       carryconnection[2],operation);
19 onebitALU A4 (result[4],carryconnection[4],setconnection[4],
       overflowconnection[4],a[4],b[4],ainvert,bnegate,constLess,
       carryconnection[3],operation);
20 onebitALU A5 (result[5],carryconnection[5],setconnection[5],
       overflowconnection[5],a[5],b[5],ainvert,bnegate,constLess,
       carryconnection[4],operation);
21 onebitALU A6 (result[6],carryconnection[6],setconnection[6],
       overflowconnection[6],a[6],b[6],ainvert,bnegate,constLess,
       carryconnection[5],operation);
22 onebitALU A7 (result[7],carryconnection[7],setconnection[7],
       overflowconnection[7],a[7],b[7],ainvert,bnegate,constLess,
       carryconnection[6],operation);
23 onebitALU A8 (result[8],carryconnection[8],setconnection[8],
       overflowconnection[8],a[8],b[8],ainvert,bnegate,constLess,
       carryconnection[7],operation);
24 onebitALU A9 (result[9],carryconnection[9],setconnection[9],
       overflowconnection[9],a[9],b[9],ainvert,bnegate,constLess,
       carryconnection[8],operation);
25 onebitALU A10 (result[10],carryconnection[10],setconnection[10],
       overflowconnection[10],a[10],b[10],ainvert,bnegate,constLess,
       carryconnection[9],operation);
26 onebitALU A11 (result[11],carryconnection[11],setconnection[11],
       overflowconnection[11],a[11],b[11],ainvert,bnegate,constLess,
       carryconnection[10],operation);
27 onebitALU A12 (result[12],carryconnection[12],setconnection[12],
       overflowconnection[12],a[12],b[12],ainvert,bnegate,constLess,
       carryconnection[11],operation);
28 onebitALU A13 (result[13],carryconnection[13],setconnection[13],
       overflowconnection[13],a[13],b[13],ainvert,bnegate,constLess,
       carryconnection[12],operation);
29 onebitALU A14 (result[14],carryconnection[14],setconnection[14],
       overflowconnection[14],a[14],b[14],ainvert,bnegate,constLess,
       carryconnection[13],operation);
30 onebitALU A15 (result[15],carryconnection[15],setconnection[15],
       overflowconnection[15],a[15],b[15],ainvert,bnegate,constLess,
       carryconnection[14],operation);
31 onebitALU A16 (result[16],carryconnection[16],setconnection[16],
```

```
    overflowconnection[16],a[16],b[16],ainvert,bnegate,constLess,
    carryconnection[15],operation);
32 onebitALU A17 (result[17],carryconnection[17],setconnection[17],
    overflowconnection[17],a[17],b[17],ainvert,bnegate,constLess,
    carryconnection[16],operation);
33 onebitALU A18 (result[18],carryconnection[18],setconnection[18],
    overflowconnection[18],a[18],b[18],ainvert,bnegate,constLess,
    carryconnection[17],operation);
34 onebitALU A19 (result[19],carryconnection[19],setconnection[19],
    overflowconnection[19],a[19],b[19],ainvert,bnegate,constLess,
    carryconnection[18],operation);
35 onebitALU A20 (result[20],carryconnection[20],setconnection[20],
    overflowconnection[20],a[20],b[20],ainvert,bnegate,constLess,
    carryconnection[19],operation);
36 onebitALU A21 (result[21],carryconnection[21],setconnection[21],
    overflowconnection[21],a[21],b[21],ainvert,bnegate,constLess,
    carryconnection[20],operation);
37 onebitALU A22 (result[22],carryconnection[22],setconnection[22],
    overflowconnection[22],a[22],b[22],ainvert,bnegate,constLess,
    carryconnection[21],operation);
38 onebitALU A23 (result[23],carryconnection[23],setconnection[23],
    overflowconnection[23],a[23],b[23],ainvert,bnegate,constLess,
    carryconnection[22],operation);
39 onebitALU A24 (result[24],carryconnection[24],setconnection[24],
    overflowconnection[24],a[24],b[24],ainvert,bnegate,constLess,
    carryconnection[23],operation);
40 onebitALU A25 (result[25],carryconnection[25],setconnection[25],
    overflowconnection[25],a[25],b[25],ainvert,bnegate,constLess,
    carryconnection[24],operation);
41 onebitALU A26 (result[26],carryconnection[26],setconnection[26],
    overflowconnection[26],a[26],b[26],ainvert,bnegate,constLess,
    carryconnection[25],operation);
42 onebitALU A27 (result[27],carryconnection[27],setconnection[27],
    overflowconnection[27],a[27],b[27],ainvert,bnegate,constLess,
    carryconnection[26],operation);
43 onebitALU A28 (result[28],carryconnection[28],setconnection[28],
    overflowconnection[28],a[28],b[28],ainvert,bnegate,constLess,
    carryconnection[27],operation);
44 onebitALU A29 (result[29],carryconnection[29],setconnection[29],
    overflowconnection[29],a[29],b[29],ainvert,bnegate,constLess,
    carryconnection[28],operation);
45 onebitALU A30 (result[30],carryconnection[30],setconnection[30],
    overflowconnection[30],a[30],b[30],ainvert,bnegate,constLess,
    carryconnection[29],operation);
46 onebitALU A31 (result[31],carryconnection[31],setconnection[31],overflow,a
    [31],b[31],ainvert,bnegate,constLess,carryconnection[30],operation);
47
48 or(invzero,result[0],result[1],result[2],result[3],result[4],result[5],
    result[6],result[7],result[8],result[9],result[10],result[11],result
    [12],result[13],result[14],result[15],result[16],result[17],result[18],
    result[19],result[20],result[21],result[22],result[23],result[24],
    result[25],result[26],result[27],result[28],result[29],result[30],
    result[31]);
49
```

```
50 not(zero,invzero);
51
52 endmodule
```

A.15 ALU Control

```
1  module ALUControl(operation,ALUOp,funcCode);
2
3  output [3:0] operation;
4  input [1:0] ALUOp;
5  input [5:0] funcCode;
6  reg [3:0] operation;
7
8  always @ (ALUOp or funcCode) begin
9      case(ALUOp)
10         2'b00:operation=4'b0010;//ADD
11         2'b01:operation=4'b0110;//SUB
12         2'b10:begin
13             case(funcCode)
14                 6'b100000:operation=4'b0010;
15                 6'b100010:operation=4'b0110;
16                 6'b100100:operation=4'b0000;
17                 6'b100101:operation=4'b0001;
18                 6'b101010:operation=4'b0111;
19             endcase
20         end
21         2'b11:operation=4'b0000;//AND
22     endcase
23 end
24
25 endmodule
```

A.16 onebitALU

```
1  module onebitALU(result,carryout,set,overflow,a,b,ainvert,binvert,less,
       carryin,operation);
2  input a,b,ainvert,binvert,less,carryin;
3  input [1:0] operation;
4  output result,carryout,set,overflow;
5  wire ainv,binv,amuxout,bmuxout,andmux0,ormux1;
6  wire medium1,medium2;
7  wire cininv,coutinv;
8  not(ainv,a);
9  not(binv,b);
10 mux2 MA2 (amuxout,a,ainv,ainvert);
11 mux2 MB2 (bmuxout,b,binv,binvert);
12 add ADD (set,carryout,amuxout,bmuxout,carryin);
13 mux4 M4 (result,andmux0,ormux1,set,less,operation);
14
15
16 and(andmux0,amuxout,bmuxout);
17 or(ormux1,amuxout,bmuxout);
18
19 not(cininv,carryin);
```

```
20 not(coutinv,carryout);
21
22 and(medium1,cininv,carryout);
23 and(medium2,carryin,coutinv);
24 or(overflow,medium1,medium2);
25 endmodule
```

A.17 mux2

```
1  module mux2(Out,In0,In1,s);
2  output Out;
3  input In0;
4  input In1;
5  input s;
6  reg Out=1'b0;
7  always @ (In0 or In1 or s)begin
8      case(s)
9      1'b0: Out=In0;
10     1'b1: Out=In1;
11     default Out=1'b0;
12     endcase
13 end
14 endmodule
```

A.18 mux4

```
1  module mux4(Out,In0,In1,In2,In3,s);
2  output Out;
3  input In0;
4  input In1;
5  input In2;
6  input In3;
7  input [1:0] s;
8  reg Out;
9  always @ (In0 or In1 or In2 or In3 or s)begin
10     case(s)
11     2'b00: Out=In0;
12     2'b01: Out=In1;
13     2'b10: Out=In2;
14     2'b11: Out=In3;
15     default Out=1'b0;
16     endcase
17 end
18 endmodule
```

A.19 add

```
1  module add(out,carry,a,b,carryin);
2  input a;
3  input b;
4  output carry;
5  input carryin;
6  output out;
7  reg out=1'b0;
```

```
reg carry;
always @ (a or b or carryin)
begin
out=(a+b+carryin)%2;
carry=(a+b+carryin)/2;
end
endmodule
```

A.20 Forwarding Unit

```
module Forward_Unit(IF_ID_RegisterRs, IF_ID_RegisterRt, ID_EX_RegisterRs,
    ID_EX_RegisterRt, EX_MEM_RegisterRd, MEM_WB_RegisterRd, EX_MEM_RegWrite
    , MEM_WB_RegWrite, ForwardA, ForwardB, ForwardC, ForwardD);

input [4:0] IF_ID_RegisterRs;
input [4:0] IF_ID_RegisterRt;
input [4:0] ID_EX_RegisterRs;
input [4:0] ID_EX_RegisterRt;
input [4:0] EX_MEM_RegisterRd;
input [4:0] MEM_WB_RegisterRd;
input EX_MEM_RegWrite;
input MEM_WB_RegWrite;
output reg  [1:0] ForwardA;
output reg  [1:0] ForwardB;
output reg  ForwardC;
output reg  ForwardD;

initial begin
    ForwardA = 2'b00;
    ForwardB = 2'b00;
    ForwardC = 1'b0;
    ForwardD = 1'b0;
end

always @(*) begin
    if (EX_MEM_RegWrite && EX_MEM_RegisterRd && EX_MEM_RegisterRd ==
        ID_EX_RegisterRs) ForwardA = 2'b10;
    else if (MEM_WB_RegWrite && MEM_WB_RegisterRd && MEM_WB_RegisterRd ==
        ID_EX_RegisterRs) ForwardA = 2'b01;
    else ForwardA=2'b00;

    if (EX_MEM_RegWrite && EX_MEM_RegisterRd && EX_MEM_RegisterRd ==
        ID_EX_RegisterRt) ForwardB = 2'b10;
    else if (MEM_WB_RegWrite && MEM_WB_RegisterRd && MEM_WB_RegisterRd ==
        ID_EX_RegisterRt) ForwardB = 2'b01;
    else ForwardB = 2'b00;

    if (EX_MEM_RegWrite && EX_MEM_RegisterRd && EX_MEM_RegisterRd ==
        IF_ID_RegisterRs) ForwardC = 1'b1; else ForwardC=1'b0;

    if (EX_MEM_RegWrite && EX_MEM_RegisterRd && EX_MEM_RegisterRd ==
        IF_ID_RegisterRt) ForwardD = 1'b1; else ForwardD=1'b0;
end

```

```
37 endmodule
```

A.21 EX/MEM Register

```
1  module EXMEM(clk, EX_Flush, EX_MemRead, EX_MemWrite, EX_MemtoReg,
       EX_RegWrite, EX_RegDst, EX_ALUResult, EX_ReadData2, MEM_MemRead,
       MEM_MemWrite, MEM_MemtoReg, MEM_RegWrite, MEM_RegDst, MEM_ALUResult,
       MEM_ReadData2);
2
3  input clk;
4  input EX_Flush;
5  input EX_MemRead;
6  input EX_MemWrite;
7  input EX_MemtoReg;
8  input EX_RegWrite;
9  input [4:0] EX_RegDst;
10 input [31:0] EX_ALUResult;
11 input [31:0] EX_ReadData2;
12 output MEM_MemRead;
13 output MEM_MemWrite;
14 output MEM_MemtoReg;
15 output MEM_RegWrite;
16 output [4:0] MEM_RegDst;
17 output [31:0] MEM_ALUResult;
18 output [31:0] MEM_ReadData2;
19 reg MEM_MemRead=1'b0;
20 reg MEM_MemWrite=1'b0;
21 reg MEM_MemtoReg=1'b0;
22 reg MEM_RegWrite=1'b0;
23 reg [4:0] MEM_RegDst=4'b0;
24 reg [31:0] MEM_ALUResult=32'b0;
25 reg [31:0] MEM_ReadData2=32'b0;
26
27 always @(posedge clk) begin
28     if (EX_Flush == 1'b1) begin
29         MEM_MemRead = 1'b0;
30         MEM_MemWrite = 1'b0;
31         MEM_MemtoReg = 1'b0;
32         MEM_RegWrite = 1'b0;
33         MEM_RegDst = 5'b0;
34         MEM_ALUResult = 32'b0;
35         MEM_ReadData2 = 32'b0;
36     end
37     else begin
38         MEM_MemRead = EX_MemRead;
39         MEM_MemWrite = EX_MemWrite;
40         MEM_MemtoReg = EX_MemtoReg;
41         MEM_RegWrite = EX_RegWrite;
42         MEM_RegDst = EX_RegDst;
43         MEM_ALUResult = EX_ALUResult;
44         MEM_ReadData2 = EX_ReadData2;
45     end
46 end
47
```

```
endmodule
```

A.22 Data Memory

```
module dataMemory(ReadData,Address,WriteData,MemWrite,MemRead,clk);

input [31:0] Address;
input [31:0] WriteData;
input MemWrite;
input MemRead;
input clk;
output [31:0] ReadData;
reg [31:0] ReadData=32'b0;
reg [31:0] storage [0:31];

initial begin
    storage[0]=32'b0;
    storage[1]=32'b0;
    storage[2]=32'b0;
    storage[3]=32'b0;
    storage[4]=32'b0;
    storage[5]=32'b0;
    storage[6]=32'b0;
    storage[7]=32'b0;
    storage[8]=32'b0;
    storage[9]=32'b0;
    storage[10]=32'b0;
    storage[11]=32'b0;
    storage[12]=32'b0;
    storage[13]=32'b0;
    storage[14]=32'b0;
    storage[15]=32'b0;
    storage[16]=32'b0;
    storage[17]=32'b0;
    storage[18]=32'b0;
    storage[19]=32'b0;
    storage[20]=32'b0;
    storage[21]=32'b0;
    storage[22]=32'b0;
    storage[23]=32'b0;
    storage[24]=32'b0;
    storage[25]=32'b0;
    storage[26]=32'b0;
    storage[27]=32'b0;
    storage[28]=32'b0;
    storage[29]=32'b0;
    storage[30]=32'b0;
    storage[31]=32'b0;
end

//For simplicity, we use max capacity of 1024 for demo.
always @ (negedge clk) begin
    if (MemWrite==1) storage[Address]<=WriteData;
    else if (MemRead==1) ReadData<=storage[Address];
```

```
51 end
52
53 endmodule
```

A.23 MEM/WB Register

```
1  module MEMWB(clk, MEM_MemtoReg, MEM_RegWrite, MEM_RegDst, MEM_ReadData,
       MEM_ALUResult, WB_MemtoReg, WB_RegWrite, WB_RegDst, WB_ReadData,
       WB_ALUResult);
2
3  input clk;
4  input MEM_MemtoReg;
5  input MEM_RegWrite;
6  input [4:0] MEM_RegDst;
7  input [31:0] MEM_ReadData;
8  input [31:0] MEM_ALUResult;
9  output WB_MemtoReg;
10 output WB_RegWrite;
11 output [4:0] WB_RegDst;
12 output [31:0] WB_ReadData;
13 output [31:0] WB_ALUResult;
14 reg WB_MemtoReg=1'b0;
15 reg WB_RegWrite=1'b0;
16 reg [4:0] WB_RegDst=4'b0;
17 reg [31:0] WB_ReadData=32'b0;
18 reg [31:0] WB_ALUResult=32'b0;
19
20 always @(posedge clk) begin
21     WB_MemtoReg = MEM_MemtoReg;
22     WB_RegWrite = MEM_RegWrite;
23     WB_RegDst = MEM_RegDst;
24     WB_ReadData = MEM_ReadData;
25     WB_ALUResult = MEM_ALUResult;
26 end
27
28 endmodule
```

A.24 Pipeline Processor

```
1  module project2pipeline(clk,PCout,VisData,Visreg);
2
3  input [4:0] Visreg;
4  output [31:0] VisData;
5  output [31:0] PCout;
6  input clk;
7  wire [31:0] NonJumpTarget;
8  wire [31:0] JumpTarget;
9  wire IDJump;
10 wire PCWrite;
11 wire [31:0] PCin;
12 wire [31:0] IFPCplusFour;
13 wire [31:0] IFinstr;
14 wire IF_IDWrite;
15 wire IDFlush;
```

```
16 wire [31:0] IDPCplusFour;
17 wire [31:0] IDinstr;
18 wire [31:0] RFOut1;
19 wire [31:0] RFOut2;
20 wire [31:0] IDReadData1;
21 wire [31:0] IDReadData2;
22 wire [1:0] ForwardA;
23 wire [1:0] ForwardB;
24 wire ForwardC;
25 wire ForwardD;
26 wire [31:0] EXExtendedImm;
27 wire [31:0] EXReadData1;
28 wire [31:0] EXReadData2;
29 wire [31:0] ALUinA;
30 wire [31:0] ALUinB;
31 wire [31:0] EXALUResult;
32 wire [31:0] MEMALUResult;
33 wire MEMMemRead;
34 wire MEMMemWrite;
35 wire [31:0] MEMMuxHOut;
36 wire [31:0] WBWriteData;
37 wire [31:0] WBReadData;
38 wire [31:0] WBALUResult;
39 wire WBMemtoReg;
40 wire stall, EXMemWrite,IDMemWrite2,IDMemWrite;
41
42 mux232 MUXA (PCin,NonJumpTarget,JumpTarget,IDJump);
43
44 PC PCA (PCout,PCin,PCWrite,clk);
45
46 reg [31:0] constFour=32'b0100;
47 Adder ADDA (IFPCplusFour,PCout,constFour);
48
49 instructionMem IMA (IFinstr,PCout);
50
51 wire BranchOrNot;
52 wire tempIDflush;
53 or(tempIDflush,BranchOrNot,IDJump);
54
55 IFID IFIDREG (clk, IF_IDWrite, tempIDflush, IFPCplusFour, IFinstr,
       IDPCplusFour, IDinstr);
56
57 //ID STAGE
58 shiftLeft26 SL26 (JumpTarget[27:0],IDinstr[25:0]);
59
60 assign JumpTarget[31:28]=IDPCplusFour[31:28];
61 wire [1:0] IDALUOp;
62 wire IDRegDst,Branch,IDMemRead,IDMemtoReg,IDALUSrc,IDRegWrite,BranchNot;
63
64 Control CONTROL (IDALUOp,IDRegDst,IDJump,Branch,IDMemRead,IDMemtoReg,
       IDMemWrite,IDALUSrc,IDRegWrite,BranchNot,IDinstr[31:26]);
65
66 wire [1:0] IDALUOp2;
67 wire IDRegDst2,IDMemRead2,IDMemtoReg2,IDALUSrc2,IDRegWrite2;
```

```
68
69  muxControl MUXB (IDALUOp2,IDRegDst2,IDALUSrc2,IDMemRead2,IDMemWrite2,
        IDRegWrite2,IDMemtoReg2,IDALUOp,IDRegDst,IDALUSrc,IDMemRead,IDMemWrite,
        IDRegWrite,IDMemtoReg,stall);
70
71  wire WBRegWrite;
72  wire [4:0] WBWriteAddr;
73
74  registers REGFILE (RFOut1,RFOut2,IDinstr[25:21],IDinstr[20:16],WBWriteAddr,
        WBWriteData,WBRegWrite,clk,VisData,Visreg);
75
76  mux232 MUXC (IDReadData1,RFOut1,MEMALUResult,ForwardC);
77
78  mux232 MUXD (IDReadData2,RFOut2,MEMALUResult,ForwardD);
79
80  wire EqualRes;
81  Equal EQUAL (EqualRes,IDReadData1,IDReadData2);
82
83  //GATES FOR BEQ AND BNE
84  wire NotEqualRes;
85  wire BeqSig,BneSig;
86
87  not(NotEqualRes,EqualRes);
88  and(BeqSig,Branch,EqualRes);
89  and(BneSig,BranchNot,NotEqualRes);
90  or(BranchOrNot,BeqSig,BneSig);
91
92  wire [31:0] IDExtendedImm;
93  signExtend SEXD (IDExtendedImm,IDinstr[15:0]);
94
95  wire [31:0] ShiftedImm;
96  shiftLeft16 SL16 (ShiftedImm,IDExtendedImm);
97
98  wire [31:0] BranchTarget;
99  Adder ADDB (BranchTarget,IDPCplusFour,ShiftedImm);
100
101 mux232 MUXF (NonJumpTarget,IFPCplusFour,BranchTarget,BranchOrNot);
102
103 wire EXMemRead,EXRegDst, EXMemtoReg, EXRegWrite,EXALUSrc;
104 wire [1:0] EXALUOp;
105 wire [4:0] EXRs, EXRt, EXRd;
106
107
108 IDEX IDEXREG (clk, IDMemRead2, IDRegDst2, IDMemWrite2, IDMemtoReg2,
        IDRegWrite2, IDReadData1, IDReadData2, IDExtendedImm, IDinstr[25:21],
        IDinstr[20:16], IDinstr[15:11], IDALUOp2, IDALUSrc2, EXMemRead,
        EXRegDst, EXMemWrite, EXMemtoReg, EXRegWrite, EXReadData1, EXReadData2,
         EXExtendedImm, EXRs, EXRt, EXRd, EXALUOp, EXALUSrc);
109
110 //EX STAGE
111 reg [31:0] constZero=32'b0;
112 mux432 MUXG (ALUinA,EXReadData1,WBWriteData,MEMALUResult,constZero,ForwardA
        );
113
```

```
114 wire [31:0] MuxHOut;
115 mux432 MUXH (MuxHOut,EXReadData2,WBWriteData,MEMALUResult,constZero,
        ForwardB);
116
117 mux232 MUXJ (ALUinB,MuxHOut,EXExtendedImm,EXALUSrc);
118
119 wire [4:0] EXDst;
120 mux25 MUXI (EXDst,EXRt,EXRd,EXRegDst);
121 wire [3:0] operation;
122 ALUControl ALUCONTROL (operation,EXALUOp,EXExtendedImm[5:0]);
123
124 wire overflow,zero;//Not used
125 ALU ALUA (EXALUResult,overflow,zero,ALUinA,ALUinB,operation[3],operation
        [2],operation[1:0]);
126
127 reg EX_Flush=1'b0; //EX_Flush const zero
128 wire MEMMemtoReg;
129 wire MEMRegWrite;
130 wire [4:0] MEMDst;
131
132 EXMEM EXMEMREG (clk, EX_Flush, EXMemRead, EXMemWrite, EXMemtoReg,
        EXRegWrite, EXDst, EXALUResult, MuxHOut, MEMMemRead, MEMMemWrite,
        MEMMemtoReg, MEMRegWrite, MEMDst, MEMALUResult, MEMMuxHOut);
133
134 //MEM STAGE
135 wire [31:0] MEMReadData;
136 dataMemory DATAMEM (MEMReadData,MEMALUResult,MEMMuxHOut,MEMMemWrite,
        MEMMemRead,clk);
137
138 MEMWB MEMWBREG (clk, MEMMemtoReg, MEMRegWrite, MEMDst, MEMReadData,
        MEMALUResult, WBMemtoReg, WBRegWrite, WBWriteAddr, WBReadData,
        WBALUResult);
139
140 //WB STAGE
141 mux232 MUXK (WBWriteData,WBALUResult,WBReadData,WBMemtoReg);
142
143 //bug fixing: for bne
144 wire HazardBranch;
145 or (HazardBranch,Branch,BranchNot);
146 Hazard_Detection_Unit HDU (IDinstr[25:21], IDinstr[20:16], EXRt,EXDst,
        MEMDst, EXMemRead, MEMMemRead, EXRegWrite, HazardBranch, PCWrite,
        IF_IDWrite, stall, IDFlush);
147
148 Forward_Unit FU (IDinstr[25:21], IDinstr[20:16], EXRs, EXRt, MEMDst,
        WBWriteAddr, MEMRegWrite, WBRegWrite, ForwardA, ForwardB, ForwardC,
        ForwardD);
149
150 endmodule
```

A.25 Implementation-related modules

A.25.1 SSD

```
1 module ssd(in,out);
```

```
input [3:0] in;
output [6:0] out;
reg [6:0] out=7'b0000000;
always @ (in)begin
case(in)
4'b0000:out<=7'b1000000;
4'b0001:out<=7'b1111001;
4'b0010:out<=7'b0100100;
4'b0011:out<=7'b0110000;
4'b0100:out<=7'b0011001;
4'b0101:out<=7'b0010010;
4'b0110:out<=7'b0000010;
4'b0111:out<=7'b1111000;
4'b1000:out<=7'b0000000;
4'b1001:out<=7'b0010000;
4'b1010:out<=7'b0001000;
4'b1011:out<=7'b0000011;
4'b1100:out<=7'b1000110;
4'b1101:out<=7'b0100001;
4'b1110:out<=7'b0000110;
4'b1111:out<=7'b0001110;
default:out<=7'b1111111;
endcase
end
endmodule
```

A.25.2 Clock Divider

```
module divider(out,reset,clk);
integer a=0;
output out;
input clk,reset;
reg out=1'b0;
always @ (posedge clk or posedge reset)
begin
if (reset==1)begin
out<=0;
a<=0;
end
else if (a==199999) begin//199999
out<=1;
a<=0;
end
else if (a==2) out<=0;

a=a+1;
end
endmodule
```

A.25.3 Ring Counter

```
module ringcount(out,reset,clk);
output [3:0] out;
input reset,clk;
```

```
reg [3:0] out=4'b1000;
always @ (posedge clk or posedge reset)
begin
    if (reset==1) out=4'b1000;
    else begin
    case(out)
    4'b0001:out=4'b1000;
    4'b1000:out=4'b0100;
    4'b0100:out=4'b0010;
    4'b0010:out=4'b0001;
    endcase
    end
end
endmodule
```

A.25.4 Buffer

```
module buffer(out,in0,in1,in2,in3,sig);
output [3:0] out;
input [3:0] in0;
input [3:0] in1;
input [3:0] in2;
input [3:0] in3;
input [3:0] sig;
reg [3:0] out;
always @ (sig or in0 or in1 or in2 or in3)begin
case(sig)
    4'b0001:out=in0;
    4'b0010:out=in1;
    4'b0100:out=in2;
    4'b1000:out=in3;
    default:out=4'b0;
endcase
end
endmodule
```

A.25.5 Mux24

```
module mux24(Out,In0,In1,s);
output Out;
input [3:0] In0;
input [3:0] In1;
input s;
reg [3:0] Out=4'b0000;
always @ (In0 or In1 or s)begin
    case(s)
    1'b0: Out=In0;
    1'b1: Out=In1;
    default Out=4'b0000;
    endcase
end
endmodule
```

A.26 Simulation

```
1  module p2pipelineSIM;
2
3  reg clk;
4  wire [31:0] NonJumpTarget;
5  wire [31:0] JumpTarget;
6  wire IDJump;
7  wire PCWrite;
8  wire [31:0] PCin;
9  wire [31:0] PCout;
10 wire [31:0] IFPCplusFour;
11 wire [31:0] IFinstr;
12 wire IF_IDWrite;
13 wire IDFlush;
14 wire [31:0] IDPCplusFour;
15 wire [31:0] IDinstr;
16 wire [31:0] RFOut1;
17 wire [31:0] RFOut2;
18 wire [31:0] IDReadData1;
19 wire [31:0] IDReadData2;
20 wire [1:0] ForwardA;
21 wire [1:0] ForwardB;
22 wire ForwardC;
23 wire ForwardD;
24 wire [31:0] EXExtendedImm;
25 wire [31:0] EXReadData1;
26 wire [31:0] EXReadData2;
27 wire [31:0] ALUinA;
28 wire [31:0] ALUinB;
29 wire [31:0] EXALUResult;
30 wire [31:0] MEMALUResult;
31 wire MEMMemRead;
32 wire MEMMemWrite;
33 wire [31:0] MEMMuxHOut;
34 wire [31:0] WBWriteData;
35 wire [31:0] WBReadData;
36 wire [31:0] WBALUResult;
37 wire WBMemtoReg;
38 wire stall, EXMemWrite,IDMemWrite2,IDMemWrite;
39
40 integer count;
41
42 project2pipeline UUT (clk, NonJumpTarget, JumpTarget, IDJump,PCWrite, PCin,
       PCout,IFPCplusFour,IFinstr,IF_IDWrite,IDFlush,IDPCplusFour,IDinstr,
       RFOut1,RFOut2,IDReadData1,IDReadData2,
43 ForwardA, ForwardB,ForwardC,ForwardD,EXExtendedImm,EXReadData1,EXReadData2,
       ALUinA,ALUinB,EXALUResult,MEMALUResult,MEMMemRead,MEMMemWrite,
       MEMMuxHOut,WBWriteData,WBReadData,WBALUResult,WBMemtoReg,stall,
       EXMemWrite,IDMemWrite2,IDMemWrite);
44
45 initial begin
46     #0 clk=0; count=1; $display("Time:%d, CLK=%d, PC=%h",0,0,00000000);
47 end
48
49 always #10 clk=~clk;
```

```
always #10
begin
    $display("Time:%d, CLK=%d, PC=%h",$time,clk,PCout);
    count=count+1;
end

initial #1000 $stop;

endmodule
```

B Textual Simulation Results

B.1 Demo Code (Bonus)

```
Time:                    0, CLK=0, PC=00000000
[$s0] = 00000000, [$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=00000000
[$s0] = 00000000, [$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=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:                   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=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, [$t9] = 00000000
Time:                  60, CLK=0, PC=0000000c
[$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, [$t9] = 00000000
Time:                  70, CLK=1, PC=0000000c
[$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, [$t9] = 00000000
Time:                  80, CLK=0, PC=00000010
[$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, [$t9] = 00000000
Time:                  90, CLK=1, PC=00000010
[$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, [$t9] = 00000000
Time:                 100, CLK=0, PC=00000014
[$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, [$t9] = 00000000
Time:                 110, CLK=1, PC=00000014
[$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, [$t9] = 00000000
Time:                 120, CLK=0, PC=00000018
[$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, [$t9] = 00000000
Time:                 130, CLK=1, PC=00000018
[$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, [$t9] = 00000000
Time:                   140, CLK=0, PC=0000001c
[$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, [$t9] = 00000000
Time:                   150, CLK=1, PC=0000001c
[$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, [$t9] = 00000000
Time:                   160, CLK=0, PC=00000020
[$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, [$t9] = 00000000
Time:                   170, CLK=1, PC=00000020
[$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, [$t9] = 00000000
Time:                   180, CLK=0, PC=00000024
[$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, [$t9] = 00000000
Time:                   190, CLK=1, PC=00000024
[$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, [$s1] = 00000000, [$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:                   210, CLK=1, 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:                   220, 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:                   230, CLK=1, 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:                   240, 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:                   250, CLK=1, 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:                   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
[$s0] = 00000037, [$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=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:                  290, CLK=1, PC=00000030
[$s0] = 00000037, [$s1] = 00000037, [$s2] = fffffffe9
[$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, [$s2] = fffffffe9
[$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
[$s0] = 00000037, [$s1] = 00000037, [$s2] = fffffffe9
[$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=00000034
[$s0] = 00000037, [$s1] = 00000037, [$s2] = fffffffe9
[$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=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:                  340, 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:                  350, CLK=1, 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:                  360, CLK=0, PC=00000038
[$s0] = 00000037, [$s1] = 00000037, [$s2] = 00000000
[$s3] = 00000000, [$s4] = 00000000, [$s5] = 00000000
```

```
256 [$s6] = 00000000, [$s7] = 00000000, [$t0] = 00000020
257 [$t1] = 00000037, [$t2] = 00000000, [$t3] = 00000000
258 [$t4] = 00000000, [$t5] = 00000000, [$t6] = 00000000
259 [$t7] = 00000000, [$t8] = 00000000, [$t9] = 00000000
260 Time:                  370, CLK=1, PC=00000038
261 [$s0] = 00000037, [$s1] = 00000037, [$s2] = 00000000
262 [$s3] = 00000000, [$s4] = 00000000, [$s5] = 00000000
263 [$s6] = 00000000, [$s7] = 00000000, [$t0] = 00000020
264 [$t1] = 00000037, [$t2] = 00000000, [$t3] = 00000000
265 [$t4] = 00000000, [$t5] = 00000000, [$t6] = 00000000
266 [$t7] = 00000000, [$t8] = 00000000, [$t9] = 00000000
267 Time:                  380, CLK=0, PC=00000038
268 [$s0] = 00000037, [$s1] = 00000037, [$s2] = 00000000
269 [$s3] = 00000000, [$s4] = 00000000, [$s5] = 00000000
270 [$s6] = 00000000, [$s7] = 00000000, [$t0] = 00000020
271 [$t1] = 00000037, [$t2] = 00000000, [$t3] = 00000000
272 [$t4] = 00000000, [$t5] = 00000000, [$t6] = 00000000
273 [$t7] = 00000000, [$t8] = 00000000, [$t9] = 00000000
274 Time:                  390, CLK=1, PC=00000038
275 [$s0] = 00000037, [$s1] = 00000037, [$s2] = 00000000
276 [$s3] = 00000020, [$s4] = 00000000, [$s5] = 00000000
277 [$s6] = 00000000, [$s7] = 00000000, [$t0] = 00000020
278 [$t1] = 00000037, [$t2] = 00000000, [$t3] = 00000000
279 [$t4] = 00000000, [$t5] = 00000000, [$t6] = 00000000
280 [$t7] = 00000000, [$t8] = 00000000, [$t9] = 00000000
281 Time:                  400, CLK=0, PC=0000003c
282 [$s0] = 00000037, [$s1] = 00000037, [$s2] = 00000000
283 [$s3] = 00000020, [$s4] = 00000000, [$s5] = 00000000
284 [$s6] = 00000000, [$s7] = 00000000, [$t0] = 00000020
285 [$t1] = 00000037, [$t2] = 00000000, [$t3] = 00000000
286 [$t4] = 00000000, [$t5] = 00000000, [$t6] = 00000000
287 [$t7] = 00000000, [$t8] = 00000000, [$t9] = 00000000
288 Time:                  410, CLK=1, PC=0000003c
289 [$s0] = 00000037, [$s1] = 00000037, [$s2] = 00000000
290 [$s3] = 00000020, [$s4] = 00000000, [$s5] = 00000000
291 [$s6] = 00000000, [$s7] = 00000000, [$t0] = 00000020
292 [$t1] = 00000037, [$t2] = 00000000, [$t3] = 00000000
293 [$t4] = 00000000, [$t5] = 00000000, [$t6] = 00000000
294 [$t7] = 00000000, [$t8] = 00000000, [$t9] = 00000000
295 Time:                  420, CLK=0, PC=00000040
296 [$s0] = 00000037, [$s1] = 00000037, [$s2] = 00000000
297 [$s3] = 00000020, [$s4] = 00000000, [$s5] = 00000000
298 [$s6] = 00000000, [$s7] = 00000000, [$t0] = 00000020
299 [$t1] = 00000037, [$t2] = 00000000, [$t3] = 00000000
300 [$t4] = 00000000, [$t5] = 00000000, [$t6] = 00000000
301 [$t7] = 00000000, [$t8] = 00000000, [$t9] = 00000000
302 Time:                  430, CLK=1, PC=00000040
303 [$s0] = 00000037, [$s1] = 00000037, [$s2] = 00000000
304 [$s3] = 00000020, [$s4] = 00000000, [$s5] = 00000000
305 [$s6] = 00000000, [$s7] = 00000000, [$t0] = 00000020
306 [$t1] = 00000037, [$t2] = 00000000, [$t3] = 00000000
307 [$t4] = 00000000, [$t5] = 00000000, [$t6] = 00000000
308 [$t7] = 00000000, [$t8] = 00000000, [$t9] = 00000000
309 Time:                  440, 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:                   450, CLK=1, 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:                   460, CLK=0, PC=00000044
[$s0] = 00000037, [$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=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:                   480, 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:                   490, CLK=1, 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:                   500, 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:                   510, CLK=1, 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
```

```
364 [$t7] = 00000000, [$t8] = 00000000, [$t9] = 00000000
365 Time:                  520, CLK=0, PC=0000003c
366 [$s0] = 00000037, [$s1] = 00000037, [$s2] = 00000000
367 [$s3] = 00000020, [$s4] = 00000001, [$s5] = 00000000
368 [$s6] = 00000000, [$s7] = 00000000, [$t0] = 00000020
369 [$t1] = 00000037, [$t2] = 00000000, [$t3] = 00000000
370 [$t4] = 00000000, [$t5] = 00000000, [$t6] = 00000000
371 [$t7] = 00000000, [$t8] = 00000000, [$t9] = 00000000
372 Time:                  530, CLK=1, PC=0000003c
373 [$s0] = 00000037, [$s1] = 00000037, [$s2] = 00000037
374 [$s3] = 00000020, [$s4] = 00000001, [$s5] = 00000000
375 [$s6] = 00000000, [$s7] = 00000000, [$t0] = 00000020
376 [$t1] = 00000037, [$t2] = 00000000, [$t3] = 00000000
377 [$t4] = 00000000, [$t5] = 00000000, [$t6] = 00000000
378 [$t7] = 00000000, [$t8] = 00000000, [$t9] = 00000000
379 Time:                  540, CLK=0, PC=00000040
380 [$s0] = 00000037, [$s1] = 00000037, [$s2] = 00000037
381 [$s3] = 00000020, [$s4] = 00000001, [$s5] = 00000000
382 [$s6] = 00000000, [$s7] = 00000000, [$t0] = 00000020
383 [$t1] = 00000037, [$t2] = 00000000, [$t3] = 00000000
384 [$t4] = 00000000, [$t5] = 00000000, [$t6] = 00000000
385 [$t7] = 00000000, [$t8] = 00000000, [$t9] = 00000000
386 Time:                  550, CLK=1, PC=00000040
387 [$s0] = 00000037, [$s1] = 00000037, [$s2] = 00000037
388 [$s3] = 00000020, [$s4] = 00000001, [$s5] = 00000000
389 [$s6] = 00000000, [$s7] = 00000000, [$t0] = 00000020
390 [$t1] = 00000037, [$t2] = 00000000, [$t3] = 00000000
391 [$t4] = 00000000, [$t5] = 00000000, [$t6] = 00000000
392 [$t7] = 00000000, [$t8] = 00000000, [$t9] = 00000000
393 Time:                  560, CLK=0, PC=00000040
394 [$s0] = 00000037, [$s1] = 00000037, [$s2] = 00000037
395 [$s3] = 00000020, [$s4] = 00000001, [$s5] = 00000000
396 [$s6] = 00000000, [$s7] = 00000000, [$t0] = 00000020
397 [$t1] = 00000037, [$t2] = 00000000, [$t3] = 00000000
398 [$t4] = 00000000, [$t5] = 00000000, [$t6] = 00000000
399 [$t7] = 00000000, [$t8] = 00000000, [$t9] = 00000000
400 Time:                  570, CLK=1, PC=00000040
401 [$s0] = 00000037, [$s1] = 00000037, [$s2] = 00000037
402 [$s3] = 00000020, [$s4] = 00000001, [$s5] = 00000000
403 [$s6] = 00000000, [$s7] = 00000000, [$t0] = 00000020
404 [$t1] = 00000037, [$t2] = 00000000, [$t3] = 00000000
405 [$t4] = 00000000, [$t5] = 00000000, [$t6] = 00000000
406 [$t7] = 00000000, [$t8] = 00000000, [$t9] = 00000000
407 Time:                  580, CLK=0, PC=0000007c
408 [$s0] = 00000037, [$s1] = 00000037, [$s2] = 00000037
409 [$s3] = 00000020, [$s4] = 00000001, [$s5] = 00000000
410 [$s6] = 00000000, [$s7] = 00000000, [$t0] = 00000020
411 [$t1] = 00000037, [$t2] = 00000000, [$t3] = 00000000
412 [$t4] = 00000000, [$t5] = 00000000, [$t6] = 00000000
413 [$t7] = 00000000, [$t8] = 00000000, [$t9] = 00000000
414 Time:                  590, CLK=1, PC=0000007c
415 [$s0] = 00000037, [$s1] = 00000037, [$s2] = 00000037
416 [$s3] = 00000020, [$s4] = 00000000, [$s5] = 00000000
417 [$s6] = 00000000, [$s7] = 00000000, [$t0] = 00000020
```

```
[$t1] = 00000037, [$t2] = 00000000, [$t3] = 00000000
[$t4] = 00000000, [$t5] = 00000000, [$t6] = 00000000
[$t7] = 00000000, [$t8] = 00000000, [$t9] = 00000000
Time:                600, CLK=0, PC=00000080
[$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, [$t9] = 00000000
Time:                610, CLK=1, PC=00000080
[$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
```

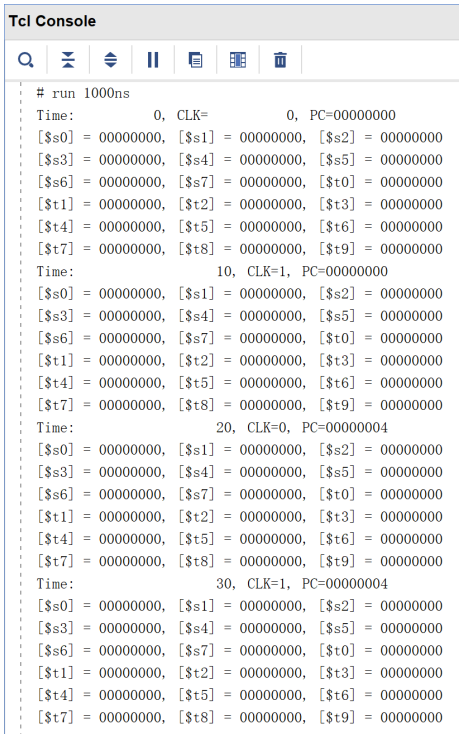
B.2 Bne Demo Code

```
Time:           0, CLK=           0, PC=00000000
[$s0] = 00000000, [$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=00000000
[$s0] = 00000000, [$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=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:                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
```

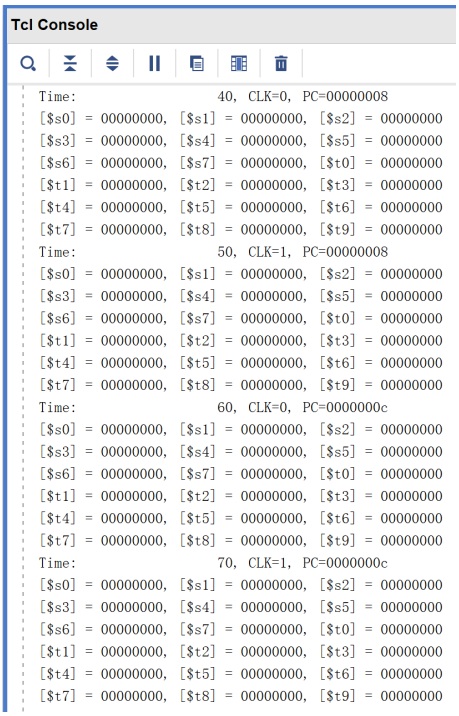
```
[$t7] = 00000000, [$t8] = 00000000, [$t9] = 00000000
Time:                   50, CLK=1, PC=00000008
[$s0] = 00000000, [$s1] = 00000000, [$s2] = 00000000
[$s3] = 00000000, [$s4] = 00000000, [$s5] = 00000000
[$s6] = 00000000, [$s7] = 00000000, [$t0] = 00000000
[$t1] = 00000000, [$t2] = 00000000, [$t3] = 00000000
[$t4] = 00000000, [$t5] = 00000000, [$t6] = 00000000
[$t7] = 00000000, [$t8] = 00000000, [$t9] = 00000000
Time:                   60, CLK=0, PC=0000000c
[$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, [$t9] = 00000000
Time:                   70, CLK=1, PC=0000000c
[$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, [$t9] = 00000000
Time:                   80, CLK=0, PC=0000000c
[$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, [$t9] = 00000000
Time:                   90, CLK=1, PC=0000000c
[$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, [$t9] = 00000000
Time:                  100, CLK=0, PC=00000048
[$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, [$t9] = 00000000
Time:                  110, CLK=1, PC=00000048
[$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
```

C Screenshot Simulation Results

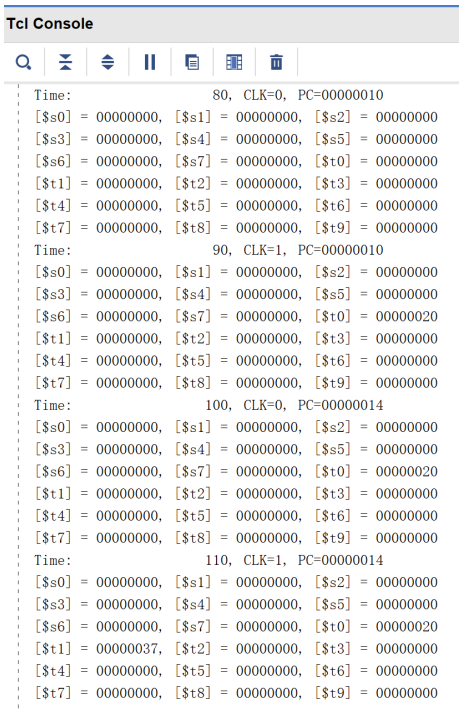
C.1 Demo Code (Bonus)



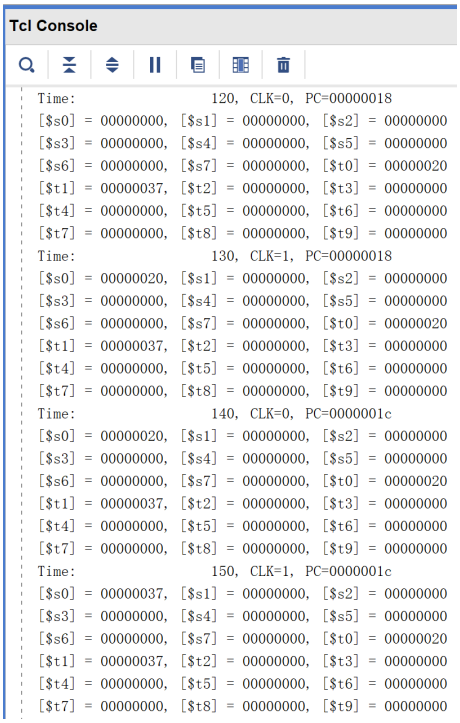
(a)



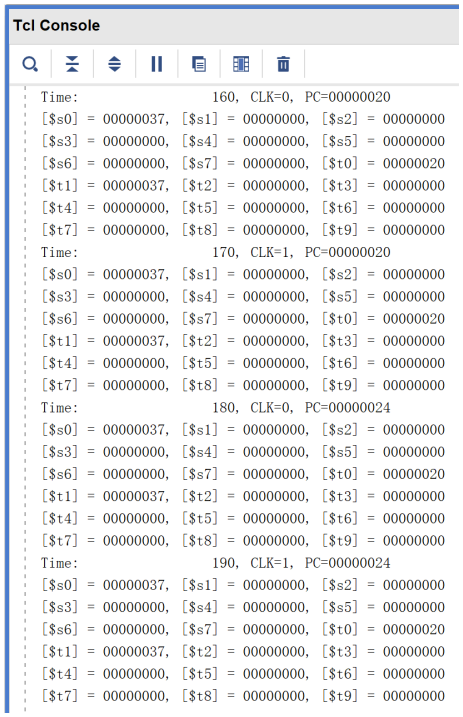
(b)


Tcl Console
Time: 80, CLK=0, PC=00000010
[$s0] = 00000000, [$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: 90, CLK=1, 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: 100, CLK=0, PC=00000014
[$s0] = 00000000, [$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: 110, CLK=1, 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

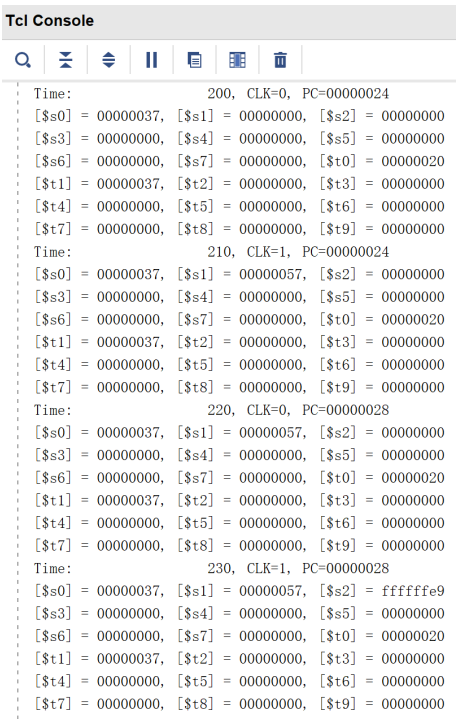
(c)


Tcl Console
Time: 120, CLK=0, PC=00000018
[$s0] = 00000000, [$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: 130, CLK=1, 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: 140, CLK=0, PC=0000001c
[$s0] = 00000020, [$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: 150, CLK=1, 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

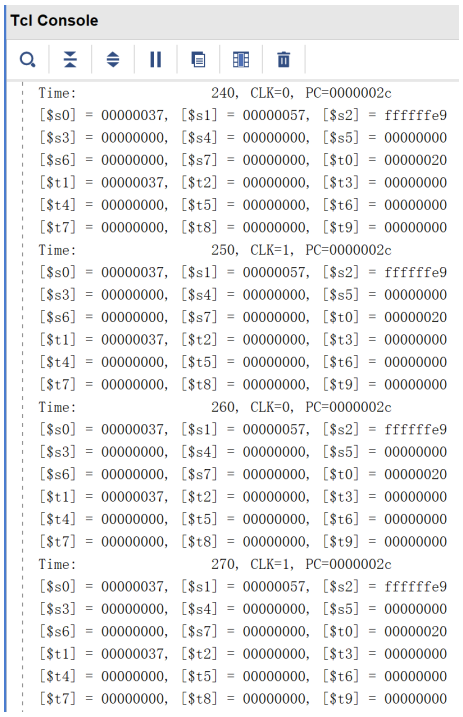
(d)


Tcl Console
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=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: 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
[$s0] = 00000037, [$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

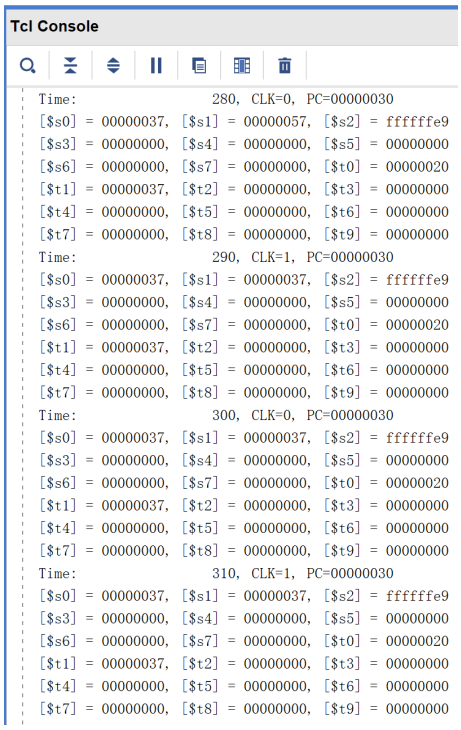
(e)


Tcl Console
Time: 200, 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: 210, CLK=1, 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: 220, 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: 230, CLK=1, PC=00000028
[$s0] = 00000037, [$s1] = 00000057, [$s2] = fffffffe9
[$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

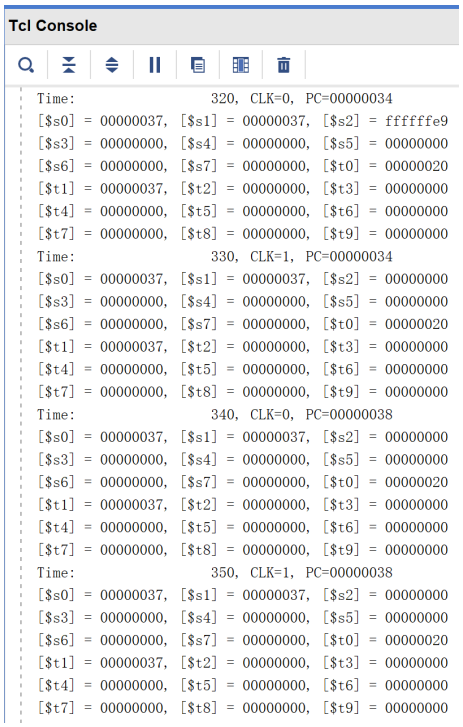
(f)


Tcl Console
Time: 240, CLK=0, PC=0000002c
[$s0] = 00000037, [$s1] = 00000057, [$s2] = fffffe9
[$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
[$s0] = 00000037, [$s1] = 00000057, [$s2] = fffffe9
[$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, [$s2] = fffffe9
[$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
[$s0] = 00000037, [$s1] = 00000057, [$s2] = fffffe9
[$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

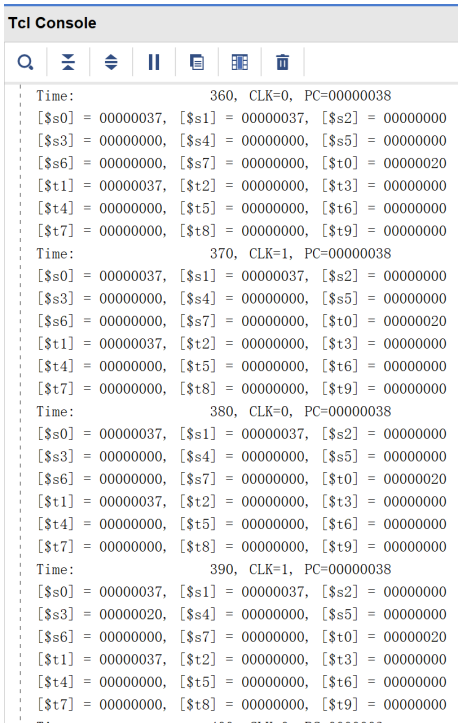
(g)


Tcl Console
Time: 280, CLK=0, PC=00000030
[$s0] = 00000037, [$s1] = 00000057, [$s2] = fffffe9
[$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
[$s0] = 00000037, [$s1] = 00000037, [$s2] = fffffe9
[$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, [$s2] = fffffe9
[$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
[$s0] = 00000037, [$s1] = 00000037, [$s2] = fffffe9
[$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

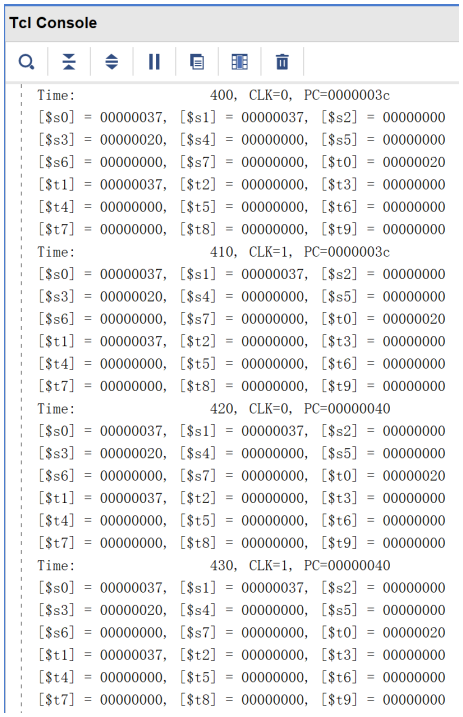
(h)


Tcl Console
Time: 320, CLK=0, PC=00000034
[$s0] = 00000037, [$s1] = 00000037, [$s2] = fffffe9
[$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=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: 340, 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: 350, CLK=1, 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

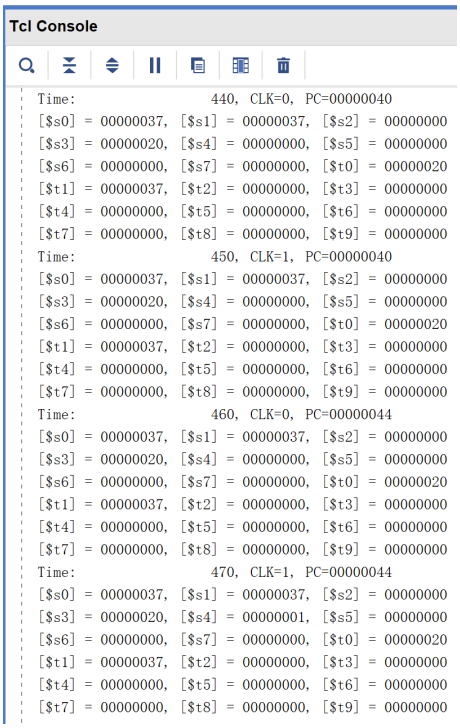
(i)


Tcl Console
Time: 360, 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: 370, CLK=1, 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: 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
[$s0] = 00000037, [$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

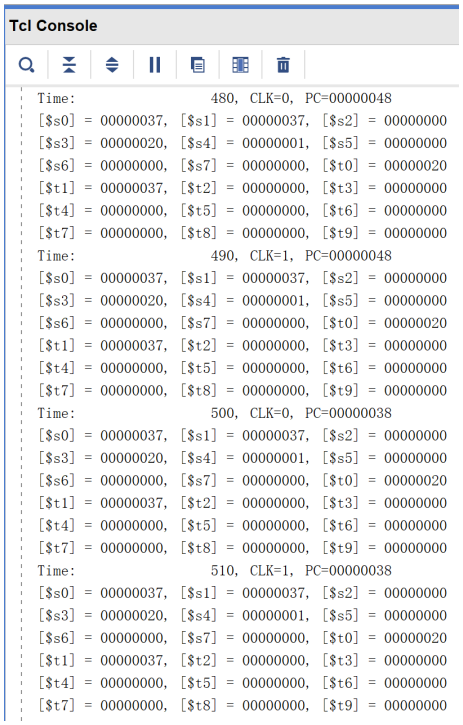
(j)


Tcl Console
Time: 400, 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: 410, CLK=1, 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: 420, 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: 430, CLK=1, 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

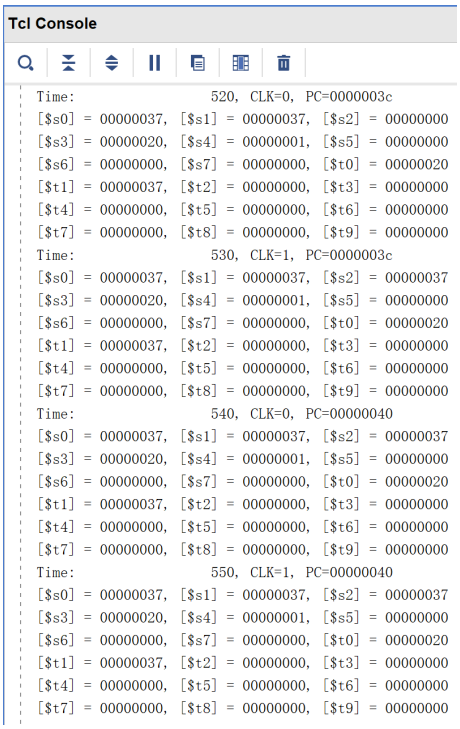
(k)


Tcl Console
Time: 440, 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: 450, CLK=1, 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: 460, CLK=0, PC=00000044
[$s0] = 00000037, [$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=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

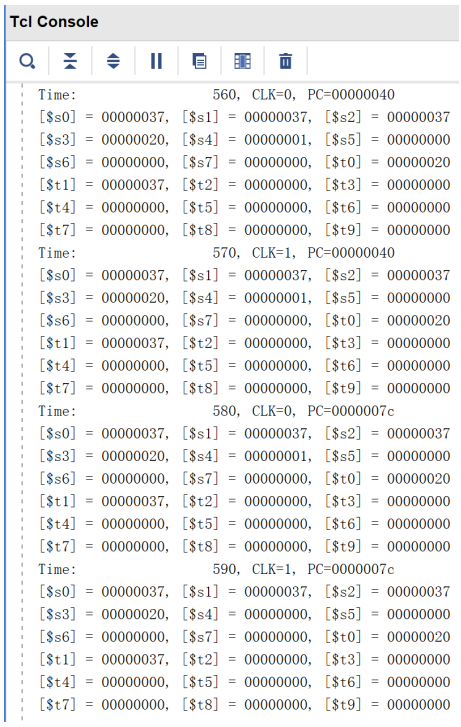
(l)


Tcl Console
Time: 480, 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: 490, CLK=1, 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: 500, 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: 510, CLK=1, 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

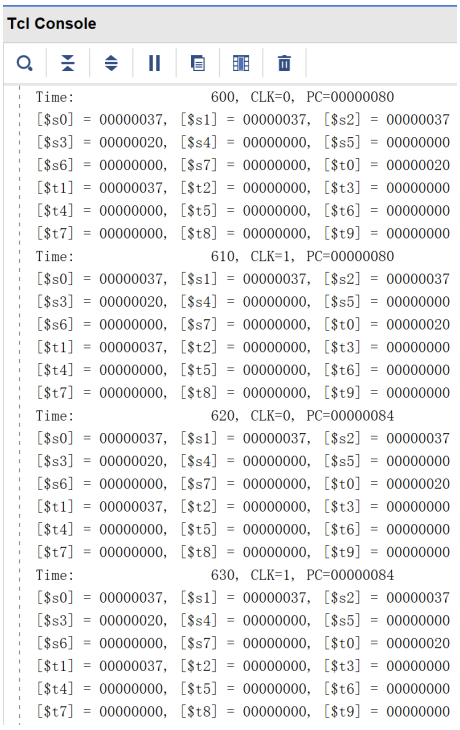
(m)


Tcl Console
Time: 520, CLK=0, PC=0000003c
[$s0] = 00000037, [$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=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: 540, 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: 550, CLK=1, 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

(n)

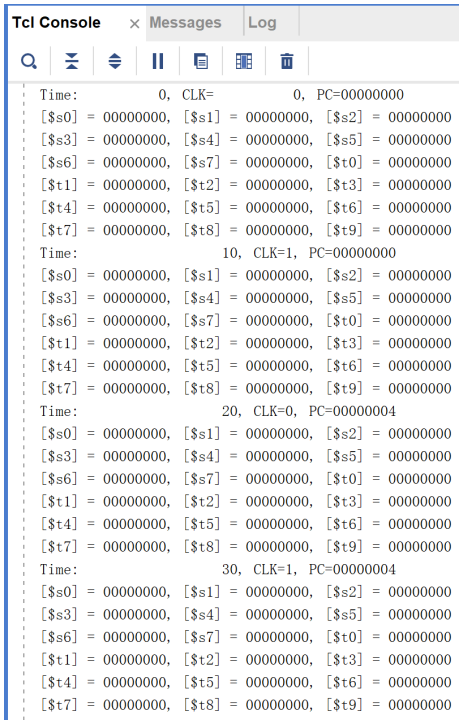


(o)

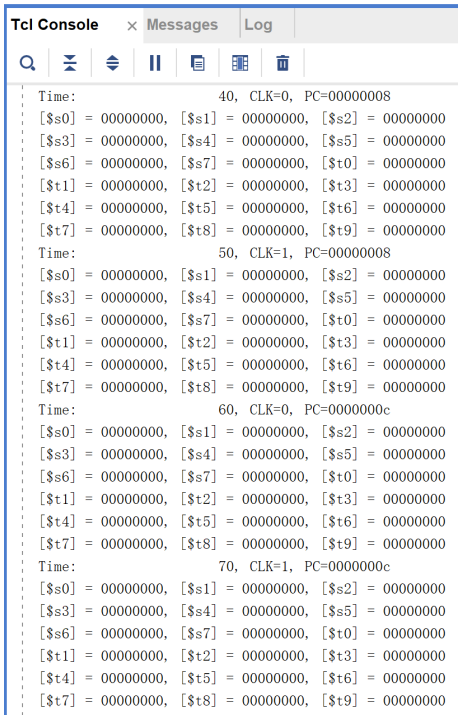


(p)

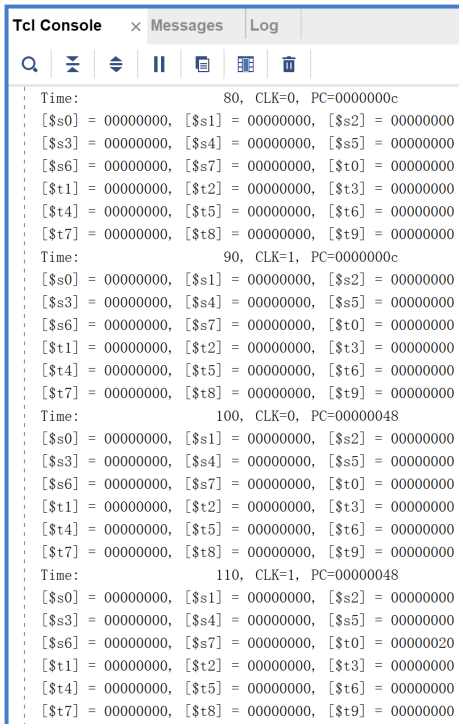
C.2 Bne Demo Code


Tcl Console
Messages
Log
Time: 0, CLK= 0, PC=00000000
[$s0] = 00000000, [$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=00000000
[$s0] = 00000000, [$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=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

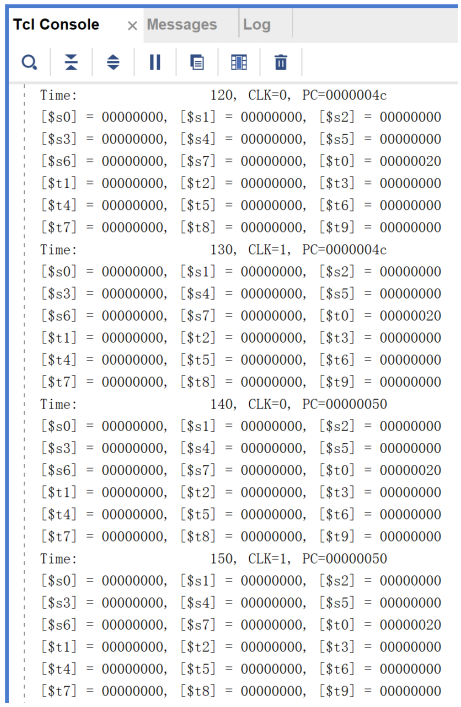
(q)


Tcl Console
Messages
Log
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=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: 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=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

(r)


Tcl Console
Messages
Log
Time: 80, 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: 90, CLK=1, 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: 100, CLK=0, PC=00000048
[$s0] = 00000000, [$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: 110, CLK=1, PC=00000048
[$s0] = 00000000, [$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

(s)


Tcl Console
Messages
Log
Time: 120, CLK=0, PC=0000004c
[$s0] = 00000000, [$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: 130, CLK=1, PC=0000004c
[$s0] = 00000000, [$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: 140, CLK=0, PC=00000050
[$s0] = 00000000, [$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: 150, CLK=1, PC=00000050
[$s0] = 00000000, [$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

(t)