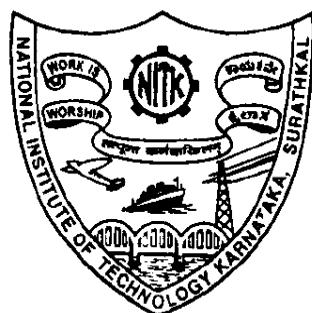


EC792 HPCA

LAB - 4



Report Submission

By

INBASEKARAN.P

201EC226

SAHIT VANKAYALA

201EE149

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA, SURATHKAL
SRINIVASNAGAR 575025 KARNATAKA INDIA

17 March 2024

Exercises.....	3
1. Adding the 4 pipeline registers between the stages.....	3
Objective:.....	3
Solution:.....	3
2. Simulating the processor.....	4
Objective:.....	4
Code:.....	4
Output Log File:.....	5
3. Stalls and Forwarding.....	7
Objective:.....	7
Code (testing ALU forwarding):.....	8
Output Log File (testing ALU forwarding):.....	8
Code (testing Memory forwarding):.....	10
Output Log File (testing Memory forwarding):.....	10
4. Integration of Hazard detection unit.....	13
Objective:.....	13
Solution:.....	14
Code:.....	14
Log File:.....	15
5. Adding the MUL Instruction.....	19
Objective:.....	19
Solution:.....	19
Code:.....	19
Output Log File:.....	20
6. Static branch prediction.....	29
Objective:.....	29
Code:.....	29
Output Log File:.....	30
7. FPGA Results.....	34
Code:.....	34
Output Log File:.....	35
FPGA output:.....	44
Resource Utilization:.....	46
RTL netlist:.....	46

Exercises

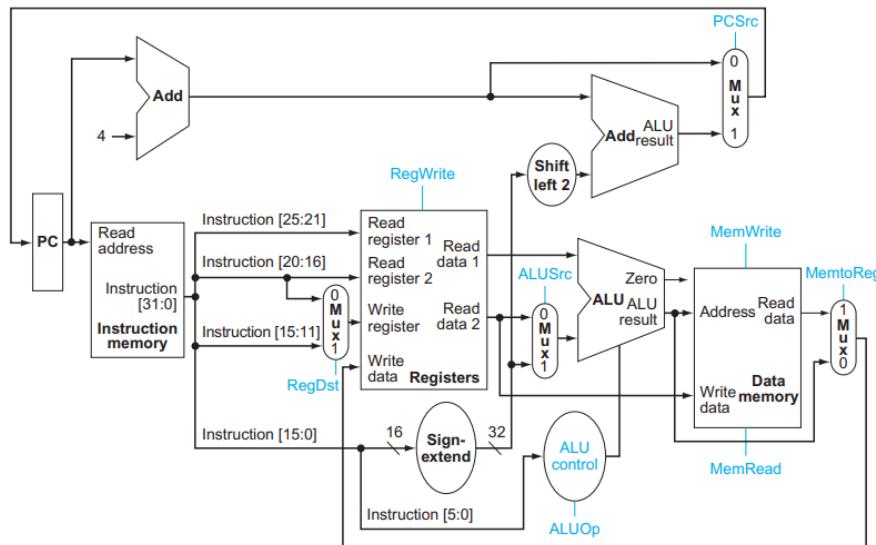
1. Adding the 4 pipeline registers between the stages

Objective:

- Add the 4 pipeline registers between the stages and connect all the data path and control signals.

Solution:

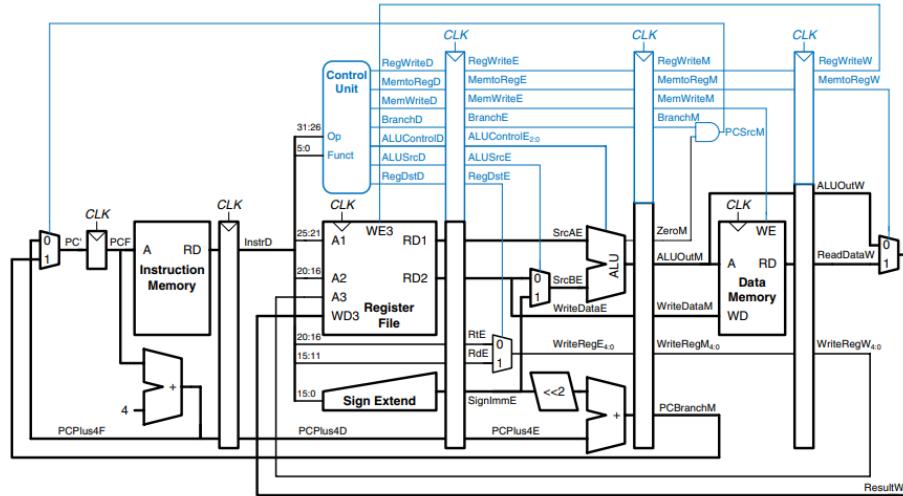
- Four modules for each stage IF/ID, ID/EX, EX/MEM, MEM/WB were created to represent the registers.
 - Initially, we have a single cycle MIPS processor(the control signals are highlighted with a blue color.)
-



Computer Organization and Design: The Hardware/software Interface

Book by David A Patterson and John L. Hennessy

- The complete MIPS 5-stage pipeline processor design with the controls looks like the following



Referred from Digital Design and Computer Architecture
Book by David M. Harris

```

90 // controller
91 assign op = instrD[31:26];
92 assign funct = instrD[5:0];
93 M_CONTROLLER controller(op, memtoreg, memwrite, branch, alusrc, regdst, regwrite, jump, aluop);
94 M_ALU_CONTROLLER aluController(funct, aluop, alucontrol);
95
96 // pipeline registers
97 > M_IF_ID_REG ifIdReg(...);
105
106
107 > M_ID_EX_REG idExReg(...);
136
137
138 > M_EX_MEM_REG exMemReg(...);
154
155
156 > M_MEM_WB_REG memWbReg(...);
170

```

Verilog code of the top module M_MIPS_CPU

NOTE:

We have followed a naming convention

- All capital letters with a M prefix for the modules
- All the variables in the module follow a camel case
- All the wires of different stages are suffixed with their respective stages.

2. Simulating the processor

Objective:

- Simulate the processor using a sequence of instructions that will not result in data or control hazards.

Code:

```
addi $s1 $zero 1
addi $s2 $zero 2
addi $s3 $zero 3
```

Output Log File:

```
current pc: xxxxxxxx
Fetched instruction 20110001
*****
Clock cycle: 1
current pc: 00000000
*****
Clock cycle: 2
Instruction 20110001 is in ID stage
Fetched instruction 20120002
stallF:0
stallD:0
flushE:0
current pc: 00000004
*****
Clock cycle: 3
Instruction 20110001 is in ID stage
Instruction 20110001 is in EX stage
Fetched instruction 20130003
Forwarded 00000001 to writedataE from EX/MEM stage
current pc: 00000008
*****
```

```
Clock cycle:          4
Instruction 20120002 is in ID stage
Instruction 20110001 is in EX stage
Instruction 20110001 is in MEM stage
current pc: 0000000c
content of $s1 = 00000001
*****
Clock cycle:          5
Instruction 20130003 is in ID stage
Instruction 20120002 is in EX stage
Instruction 20110001 is in MEM stage
Instruction 20110001 is in WB stage
stallF:x
stallD:x
flushE:x
current pc: 00000010
content of $s1 = 00000001
*****
Clock cycle:          6
Instruction 20130003 is in EX stage
Instruction 20120002 is in MEM stage
Instruction 20110001 is in WB stage
current pc: 00000014
content of $s2 = 00000002
*****
Clock cycle:          7
Instruction 20130003 is in MEM stage
Instruction 20120002 is in WB stage
current pc: 00000018
content of $s3 = 00000003
*****
Clock cycle:          8
```

Instruction 20130003 is in WB stage

current pc: 0000001c

Clock cycle: 9

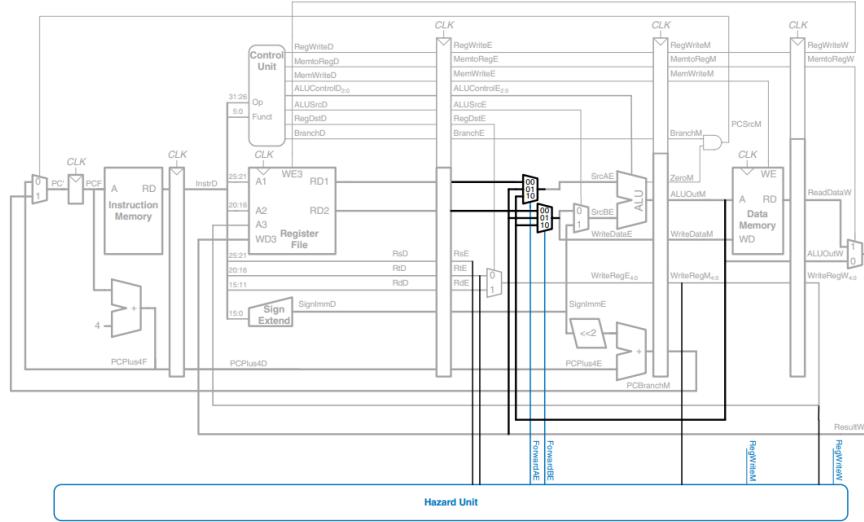
current pc: 00000020

In order to make the debugging process easier, the important control signals and write_reg values from the register-file along were displayed during the simulation using the \$display system command.

3. Stalls and Forwarding

Objective:

- Include forwarding logic, simulate using a sequence of instructions with data hazards that the forwarding logic Solution can handle:
 - Forwarding in MIPS allows data to be transferred directly between pipeline stages, reducing stalls caused by dependencies, enhancing performance, and avoiding hazards.
 - To Include forwarding, we add two 4 X 2 Muxes(with control signals Forward A and Forward B) and check for a specific condition where forwarding may be required. Specifically,
 - (EX/MEM.RegWrite) and (EX/MEM.RegisterRd ≠ 0) and (EX/MEM.RegisterRd = ID/EX.RegisterRs) ForwardA = 10 EX Hazard
 - (EX/MEM.RegWrite and (EX/MEM.RegisterRd ≠ 0) and (EX/MEM.RegisterRd = ID/EX.RegisterRt)) ForwardB = 10 EX Hazard
 - (MEM/WB.RegWrite and (MEM/WB.RegisterRd ≠ 0) and (MEM/WB.RegisterRd = ID/EX.RegisterRs)) and not(EX/MEM.RegWrite and (EX/MEM.RegisterRd ≠ 0) and (EX/MEM.RegisterRd ≠ ID/EX.RegisterRs)) ForwardA = 01 MEM Hazard
 - (MEM/WB.RegWrite and (MEM/WB.RegisterRd ≠ 0) and (MEM/WB.RegisterRd = ID/EX.RegisterRt)) and not(EX/MEM.RegWrite and (EX/MEM.RegisterRd ≠ 0) and (EX/MEM.RegisterRd ≠ ID/EX.RegisterRt)) ForwardB = 01 MEM Hazard



Referred from Digital Design and Computer Architecture
Book by David M. Harris

Code (testing ALU forwarding):

```
addi $s1, $zero, 0x1
addi $s2, $s1, 0x2
and $s3, $s1, $s2 // $s3 should have 3 in the end
```

Output Log File (testing ALU forwarding):

```
current pc: xxxxxxxx
Fetched instruction 20110001
*****
Clock cycle: 1
current pc: 00000000
*****
Clock cycle: 2
Instruction 20110001 is in ID stage
Fetched instruction 20120002
stallF:0
stallD:0
flushE:0
```

```
current pc: 00000004
*****
Clock cycle:      3
Instruction 20110001 is in ID stage
Instruction 20110001 is in EX stage
Fetched instruction 20130003
Forwarded 00000001 to writedataE from EX/MEM stage
current pc: 00000008
*****
Clock cycle:      4
Instruction 20120002 is in ID stage
Instruction 20110001 is in EX stage
Instruction 20110001 is in MEM stage
current pc: 0000000c
content of $s1 = 00000001
*****
Clock cycle:      5
Instruction 20130003 is in ID stage
Instruction 20120002 is in EX stage
Instruction 20110001 is in MEM stage
Instruction 20110001 is in WB stage
stallF:x
stallD:x
flushE:x
current pc: 00000010
content of $s1 = 00000001
*****
Clock cycle:      6
Instruction 20130003 is in EX stage
Instruction 20120002 is in MEM stage
Instruction 20110001 is in WB stage
current pc: 00000014
```

```

content of $s2 = 00000002
*****
Clock cycle:      7
Instruction 20130003 is in MEM stage
Instruction 20120002 is in WB stage
current pc: 00000018
content of $s3 = 00000003
*****
Clock cycle:      8
Instruction 20130003 is in WB stage
current pc: 0000001c
*****
Clock cycle:      9
current pc: 00000020
*****

```

Additionally we added forwarding functionality for the load word instructions also, the functionality was tested with the following code.

Code (testing Memory forwarding):

```

addi $s1 $zero 0x1
sw $s1 0x4($zero)
addi $s2 $s1 0x2
lw $s0 0x4($zero)
add $t0 $s0 $s1
add $t1 $s2 $s0
add $t2 $s0 $s2

```

Output Log File (testing Memory forwarding):

```

current pc: xxxxxxxx
Fetched instruction 20110001
*****
Clock cycle:      1

```

```
current pc: 00000000
*****
Clock cycle:      2
Instruction 20110001 is in ID stage
Fetched instruction ac110004
stallF:0
stallD:0
flushE:0
current pc: 00000004
*****
Clock cycle:      3
Instruction 20110001 is in ID stage
Instruction 20110001 is in EX stage
Fetched instruction 22320002
Forwarded 00000001 to writedataE from EX/MEM stage
current pc: 00000008
*****
Clock cycle:      4
Instruction ac110004 is in ID stage
Instruction 20110001 is in EX stage
Instruction 20110001 is in MEM stage
Fetched instruction 8c100004
current pc: 0000000c
content of $s1 = 00000001
*****
Clock cycle:      5
Instruction 22320002 is in ID stage
Instruction ac110004 is in EX stage
Instruction 20110001 is in MEM stage
Instruction 20110001 is in WB stage
Fetched instruction 02114020
Forwarded 00000001 to srcaE from MEM/WB stage
```

```
current pc: 00000010
content of $s1 = 00000001
*****
Clock cycle:      6
address 00000004 now has data 00000001
Instruction 8c100004 is in ID stage
Instruction 22320002 is in EX stage
Instruction ac110004 is in MEM stage
Instruction 20110001 is in WB stage
Fetched instruction 02504820
lwstall=1
stallF:1
stallD:1
flushE:1
current pc: 00000014
*****
Clock cycle:      7
Instruction 22320002 is in MEM stage
Instruction ac110004 is in WB stage
stallF:0
stallD:0
flushE:0
current pc: 00000014
content of $s2 = 00000003
*****
Clock cycle:      8
Instruction 02114020 is in ID stage
Instruction 8c100004 is in MEM stage
Instruction 22320002 is in WB stage
Fetched instruction 02125020
Forwarded 00000001 to srcaE from MEM/WB stage
current pc: 00000018
```

```
content of $s0 = 00000001
*****
Clock cycle:      9
Instruction 02504820 is in ID stage
Instruction 02114020 is in EX stage
Instruction 8c100004 is in WB stage
current pc: 0000001c
*****

Clock cycle:      10
Instruction 02125020 is in ID stage
Instruction 02504820 is in EX stage
Instruction 02114020 is in MEM stage
stallF:x
stallD:x
flushE:x
current pc: 00000020
content of $t0 = 00000002
*****
```

```
Clock cycle:      11
Instruction 02125020 is in EX stage
Instruction 02504820 is in MEM stage
Instruction 02114020 is in WB stage
current pc: 00000024
content of $t1 = 00000004
*****
```

```
Clock cycle:      12
Instruction 02125020 is in MEM stage
Instruction 02504820 is in WB stage
current pc: 00000028
content of $t2 = 00000004
*****
```

```
Clock cycle:      13
```

Instruction 02125020 is in WB stage

current pc: 0000002c

Clock cycle: 14

current pc: 00000030

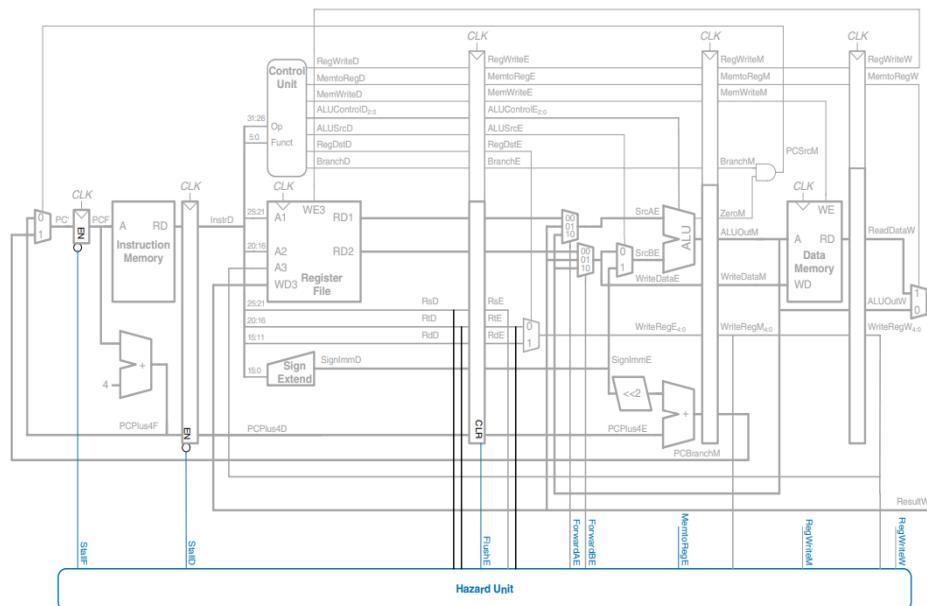
4. Integration of Hazard detection unit

Objective:

- Include hazard detection logic, simulate using a sequence of instructions with
 - (a) control hazards that the hazard detection logic can handle
 - and (b) both data and control hazards

Solution:

- (a) Hazard detection in MIPS processors identifies and resolves conflicts arising from data dependencies, ensuring accurate instruction execution and maintaining pipeline efficiency. The implementation of the same can be visualized as follows.



Referred from Digital Design and Computer Architecture

Book by David M. Harris

- Stall is introduced into the pipeline if
(ID/EX.MemRead and ((ID/EX.RegisterRt = IF/ID.RegisterRs) or (ID/EX.RegisterRt = IF/ID.RegisterRt))).

- If the instruction in the ID stage is stalled, then the instruction in the IF stage must also be stalled; otherwise, we would lose the fetched instruction. Deasserting all nine control signals (setting them to 0) in the EX, MEM, and WB stages will create a “do nothing” or nop instruction. By identifying the hazard in the ID stage, we can insert a bubble into the pipeline by changing the EX, MEM, and WB control fields of the ID/EX pipeline register to 0. These benign control values are percolated forward at each clock cycle with the proper effect: no registers or memories are written if the control values are all 0.

Code:

```

addi $t0 $zero 0x1
addi $s2 $zero 0x2
sub $t1 $s2 $t0 // $t1 should have 1
beq $t0 $s2 0x4 // not taken
beq $t0 $t1 0x5 // taken
addi $s0 $zero 0x1
addi $s0 $s0 0x1
addi $s0 $s0 0x1
addi $s0 $s0 0x1
addi $s0 $s0 0x1
sw $s2 0x8($zero) // beq will go to here
slt $s1 $t0 $s2
lw $s3 0x8($zero)
add $s4 $s3 $s1

```

Log File:

```

current pc: xxxxxxxx
Fetched instruction 20080001
*****
Clock cycle: 1
current pc: 00000000
*****
Clock cycle: 2
Instruction 20080001 is in ID stage
Fetched instruction 20120002

```

```

stallF:0
stallD:0
flushE:0
current pc: 00000004
*****
Clock cycle:      3
Instruction 20080001 is in ID stage
Instruction 20080001 is in EX stage
Fetched instruction 02484822
Forwarded 00000001 to writedataE from EX/MEM stage
current pc: 00000008
*****
Clock cycle:      4
Instruction 20120002 is in ID stage
Instruction 20080001 is in EX stage
Instruction 20080001 is in MEM stage
Fetched instruction 11120004
current pc: 0000000c
content of $t0 = 00000001
*****
Clock cycle:      5
Instruction 02484822 is in ID stage
Instruction 20120002 is in EX stage
Instruction 20080001 is in MEM stage
Instruction 20080001 is in WB stage
Fetched instruction 11090005
Forwarded 00000001 to writedataE from MEM/WB stage
Forwarded 00000002 to srcaE from EX/MEM stage
Branch not taken
current pc: 00000010
content of $t0 = 00000001
*****

```

```
Clock cycle:      6
Instruction 11120004 is in ID stage
Instruction 02484822 is in EX stage
Instruction 20120002 is in MEM stage
Instruction 20080001 is in WB stage
Fetched instruction 20100001
Branch is taken! Squashing instruction 20100001
current pc: 00000014
content of $s2 = 00000002
*****
Clock cycle:      7
Instruction 11120004 is in EX stage
Instruction 02484822 is in MEM stage
Instruction 20120002 is in WB stage
Fetched instruction ac120008
current pc: 00000028
content of $t1 = 00000001
*****
Clock cycle:      8
Instruction 11090005 is in EX stage
Instruction 11120004 is in MEM stage
Instruction 02484822 is in WB stage
Fetched instruction 0112882a
current pc: 0000002c
*****
Clock cycle:      9
Instruction ac120008 is in ID stage
Instruction 11090005 is in MEM stage
Instruction 11120004 is in WB stage
Fetched instruction 8c130008
current pc: 00000030
*****
```

```
Clock cycle:      10
Instruction 0112882a is in ID stage
Instruction ac120008 is in EX stage
Instruction 11090005 is in WB stage
Fetched instruction 0271a020
current pc: 00000034
no data written to register file
*****
Clock cycle:      11
address 00000008 now has data 00000002
Instruction 8c130008 is in ID stage
Instruction 0112882a is in EX stage
Instruction ac120008 is in MEM stage
lwstall=1
stallF:1
stallD:1
flushE:1
current pc: 00000038
*****
Clock cycle:      12
Instruction 0112882a is in MEM stage
Instruction ac120008 is in WB stage
stallF:0
stallD:0
flushE:0
current pc: 00000038
content of $s1 = 00000001
*****
Clock cycle:      13
Instruction 0271a020 is in ID stage
Instruction 8c130008 is in MEM stage
Instruction 0112882a is in WB stage
```

```

Forwarded 00000002 to srcaE from MEM/WB stage
stallF:x
stallD:x
flushE:x
current pc: 0000003c
content of $s3 = 00000002
*****
Clock cycle:      14
Instruction 0271a020 is in EX stage
Instruction 8c130008 is in WB stage
current pc: 00000040
*****
Clock cycle:      15
Instruction 0271a020 is in MEM stage
current pc: 00000044
content of $s4 = 00000003
*****
Clock cycle:      16
Instruction 0271a020 is in WB stage
current pc: 00000048
*****
Clock cycle:      17
current pc: 0000004c
*****

```

5. Adding the MUL Instruction

Objective:

- Add MUL instruction by adding pipelined integer multiply execution unit.

Solution:

- The 32-bit Array Multiplier, which was created in Lab 6 of DDFPGA, has been integrated into the ALU to perform multiplication.

- The MUL instruction follows the MIPS ISA, as referenced in this PDF (<https://s3-eu-west-1.amazonaws.com/downloads-mips/documents/MD00086-2B-MIPS32BIS-AFP-6.06.pdf>) on page number 299.

To test the multiplication functionality we calculated the factorial of 6, and the result 720 is correctly stored in the register \$s0 in the 29th clock cycle.

Code:

```
.text
INIT:
    addi $t0, $zero, 0x1
    addi $s1, $zero, 0x1
ITER_1:
    mul $s1, $s1 , $t0
    addi $t0, $t0, 0x1
ITER_2:
    mul $s1, $s1 , $t0
    addi $t0, $t0, 0x1
ITER_3:
    mul $s1, $s1 , $t0
    addi $t0, $t0, 0x1
ITER_4:
    mul $s1, $s1 , $t0
    addi $t0, $t0, 0x1
ITER_5:
    mul $s1, $s1 , $t0
    addi $t0, $t0, 0x1
ITER_6:
    mul $s1, $s1 , $t0
    addi $t0, $t0, 0x1
```

Output Log File:

```
current pc: xxxxxxxx
Fetched instruction 20080001
```

```
*****
Clock cycle:          1
current pc: 00000000
*****
Clock cycle:          2
Instruction 20080001 is in ID stage
Fetched instruction 20110001
stallF:0
stallD:0
flushE:0
current pc: 00000004
*****
Clock cycle:          3
Instruction 20080001 is in ID stage
Instruction 20080001 is in EX stage
Fetched instruction 72288802
Forwarded 00000001 to writedataE from EX/MEM stage
current pc: 00000008
*****
Clock cycle:          4
Instruction 20110001 is in ID stage
Instruction 20080001 is in EX stage
Instruction 20080001 is in MEM stage
Fetched instruction 20080001
current pc: 0000000c
content of $t0 = 00000001
*****
Clock cycle:          5
Instruction 72288802 is in ID stage
Instruction 20110001 is in EX stage
Instruction 20080001 is in MEM stage
Instruction 20080001 is in WB stage
```

```
Fetched instruction 72288802
Forwarded 00000001 to writedataE from MEM/WB stage
Forwarded 00000001 to srcaE from EX/MEM stage
current pc: 00000010
content of $t0 = 00000001
*****
Clock cycle: 6
Instruction 20080001 is in ID stage
Instruction 72288802 is in EX stage
Instruction 20110001 is in MEM stage
Instruction 20080001 is in WB stage
Fetched instruction 20080001
current pc: 00000014
content of $s1 = 00000001
*****
Clock cycle: 7
Instruction 72288802 is in ID stage
Instruction 20080001 is in EX stage
Instruction 72288802 is in MEM stage
Instruction 20110001 is in WB stage
Fetched instruction 72288802
Forwarded 00000001 to writedataE from EX/MEM stage
Forwarded 00000001 to srcaE from MEM/WB stage
current pc: 00000018
content of $s1 = 00000001
*****
Clock cycle: 8
Instruction 20080001 is in ID stage
Instruction 72288802 is in EX stage
Instruction 20080001 is in MEM stage
Instruction 72288802 is in WB stage
Fetched instruction 20080001
```

```
Forwarded 00000001 to writedataE from MEM/WB stage
current pc: 0000001c
content of $t0 = 00000001
*****
Clock cycle: 9
Instruction 72288802 is in ID stage
Instruction 20080001 is in EX stage
Instruction 72288802 is in MEM stage
Instruction 20080001 is in WB stage
Fetched instruction 72288802
Forwarded 00000001 to writedataE from EX/MEM stage
Forwarded 00000001 to srcaE from MEM/WB stage
current pc: 00000020
content of $s1 = 00000001
*****
Clock cycle: 10
Instruction 20080001 is in ID stage
Instruction 72288802 is in EX stage
Instruction 20080001 is in MEM stage
Instruction 72288802 is in WB stage
Fetched instruction 20080001
Forwarded 00000001 to writedataE from MEM/WB stage
current pc: 00000024
content of $t0 = 00000001
*****
Clock cycle: 11
Instruction 72288802 is in ID stage
Instruction 20080001 is in EX stage
Instruction 72288802 is in MEM stage
Instruction 20080001 is in WB stage
Fetched instruction 72288802
Forwarded 00000001 to writedataE from EX/MEM stage
```

```
Forwarded 00000001 to srcaE from MEM/WB stage
current pc: 00000028
content of $s1 = 00000001
*****
Clock cycle: 12
Instruction 20080001 is in ID stage
Instruction 72288802 is in EX stage
Instruction 20080001 is in MEM stage
Instruction 72288802 is in WB stage
Fetched instruction 20080001
Forwarded 00000001 to writedataE from MEM/WB stage
current pc: 0000002c
content of $t0 = 00000001
*****
Clock cycle: 13
Instruction 72288802 is in ID stage
Instruction 20080001 is in EX stage
Instruction 72288802 is in MEM stage
Instruction 20080001 is in WB stage
Forwarded 00000001 to writedataE from EX/MEM stage
Forwarded 00000001 to srcaE from MEM/WB stage
current pc: 00000030
content of $s1 = 00000001
*****
Clock cycle: 14
Instruction 20080001 is in ID stage
Instruction 72288802 is in EX stage
Instruction 20080001 is in MEM stage
Instruction 72288802 is in WB stage
Fetched instruction 20110001
Forwarded 00000001 to writedataE from MEM/WB stage
current pc: 00000034
```

```
content of $t0 = 00000001
*****
Clock cycle:      15
Instruction 20080001 is in ID stage
Instruction 20080001 is in EX stage
Instruction 72288802 is in MEM stage
Instruction 20080001 is in WB stage
Fetched instruction 72288802
Forwarded 00000001 to writedataE from EX/MEM stage
current pc: 00000038
content of $s1 = 00000001
*****
Clock cycle:      16
Instruction 20110001 is in ID stage
Instruction 20080001 is in EX stage
Instruction 20080001 is in MEM stage
Instruction 72288802 is in WB stage
Fetched instruction 21080001
current pc: 0000003c
content of $t0 = 00000001
*****
Clock cycle:      17
Instruction 72288802 is in ID stage
Instruction 20110001 is in EX stage
Instruction 20080001 is in MEM stage
Instruction 20080001 is in WB stage
Fetched instruction 72288802
Forwarded 00000001 to writedataE from MEM/WB stage
Forwarded 00000001 to srcaE from EX/MEM stage
current pc: 00000040
content of $t0 = 00000001
*****
```

```
Clock cycle:      18
Instruction 21080001 is in ID stage
Instruction 72288802 is in EX stage
Instruction 20110001 is in MEM stage
Instruction 20080001 is in WB stage
Fetched instruction 21080001
current pc: 00000044
content of $s1 = 00000001
*****
Clock cycle:      19
Instruction 72288802 is in ID stage
Instruction 21080001 is in EX stage
Instruction 72288802 is in MEM stage
Instruction 20110001 is in WB stage
Fetched instruction 72288802
Forwarded 00000002 to writedataE from EX/MEM stage
Forwarded 00000001 to srcaE from MEM/WB stage
current pc: 00000048
content of $s1 = 00000001
*****
Clock cycle:      20
Instruction 21080001 is in ID stage
Instruction 72288802 is in EX stage
Instruction 21080001 is in MEM stage
Instruction 72288802 is in WB stage
Fetched instruction 21080001
Forwarded 00000002 to writedataE from MEM/WB stage
current pc: 0000004c
content of $t0 = 00000002
*****
Clock cycle:      21
Instruction 72288802 is in ID stage
```

```
Instruction 21080001 is in EX stage
Instruction 72288802 is in MEM stage
Instruction 21080001 is in WB stage
Fetched instruction 72288802
Forwarded 00000003 to writedataE from EX/MEM stage
current pc: 00000050
content of $s1 = 00000002
*****
Clock cycle: 22
Instruction 21080001 is in ID stage
Instruction 72288802 is in EX stage
Instruction 21080001 is in MEM stage
Instruction 72288802 is in WB stage
Fetched instruction 21080001
Forwarded 00000003 to writedataE from MEM/WB stage
current pc: 00000054
content of $t0 = 00000003
*****
Clock cycle: 23
Instruction 72288802 is in ID stage
Instruction 21080001 is in EX stage
Instruction 72288802 is in MEM stage
Instruction 21080001 is in WB stage
Fetched instruction 72288802
Forwarded 00000004 to writedataE from EX/MEM stage
current pc: 00000058
content of $s1 = 00000006
*****
Clock cycle: 24
Instruction 21080001 is in ID stage
Instruction 72288802 is in EX stage
Instruction 21080001 is in MEM stage
```

```
Instruction 72288802 is in WB stage
Fetched instruction 21080001
Forwarded 00000004 to writedataE from MEM/WB stage
current pc: 0000005c
content of $t0 = 00000004
*****
Clock cycle: 25
Instruction 72288802 is in ID stage
Instruction 21080001 is in EX stage
Instruction 72288802 is in MEM stage
Instruction 21080001 is in WB stage
Fetched instruction 72288802
Forwarded 00000005 to writedataE from EX/MEM stage
current pc: 00000060
content of $s1 = 00000018
*****
Clock cycle: 26
Instruction 21080001 is in ID stage
Instruction 72288802 is in EX stage
Instruction 21080001 is in MEM stage
Instruction 72288802 is in WB stage
Fetched instruction 21080001
Forwarded 00000005 to writedataE from MEM/WB stage
current pc: 00000064
content of $t0 = 00000005
*****
Clock cycle: 27
Instruction 72288802 is in ID stage
Instruction 21080001 is in EX stage
Instruction 72288802 is in MEM stage
Instruction 21080001 is in WB stage
Forwarded 00000006 to writedataE from EX/MEM stage
```

```
current pc: 00000068
content of $s1 = 00000078
*****
Clock cycle: 28
Instruction 21080001 is in ID stage
Instruction 72288802 is in EX stage
Instruction 21080001 is in MEM stage
Instruction 72288802 is in WB stage
Forwarded 00000006 to writedataE from MEM/WB stage
stallF:x
stallD:x
flushE:x
current pc: 0000006c
content of $t0 = 00000006
*****
Clock cycle: 29
Instruction 21080001 is in EX stage
Instruction 72288802 is in MEM stage
Instruction 21080001 is in WB stage
current pc: 00000070
content of $s1 = 000002d0
*****
Clock cycle: 30
Instruction 21080001 is in MEM stage
Instruction 72288802 is in WB stage
current pc: 00000074
content of $t0 = 00000007
*****
Clock cycle: 31
Instruction 21080001 is in WB stage
current pc: 00000078
*****
```

```
Clock cycle:      32
current pc: 0000007c
*****
```

6. Static branch prediction

Objective:

- Adding Static branch prediction.
- Chosen Static branch prediction: Assume branch is not taken always.

Code:

```
addi $t0 $zero 0x1
addi $s2 $zero 0x2
sub $t1 $s2 $t0 // $t1 should have 1
beq $t0 $s2 0x4 // not taken
beq $t0 $t1 0x5 // taken
addi $s0 $zero 0x1
addi $s0 $s0 0x1
addi $s0 $s0 0x1
addi $s0 $s0 0x1
addi $s0 $s0 0x1
sw $s2 0x8($zero) // beq will go to here
slt $s1 $t0 $s2
lw $s3 0x8($zero)
add $s4 $s3 $s1
```

The following code tests the branch instruction, when the 5th instruction (beq \$t0 \$t1 0x5 // taken) is executed, we can see from the log files that the next instruction gets squashed as the branch prediction was wrong. From the log files, we can see that at **cycle 6** the next instruction (addi \$s0 \$zero 0x1) is being flushed.

Output Log File:

```
current pc: xxxxxxxx
Fetched instruction 20080001
```

```
*****
Clock cycle:          1
current pc: 00000000
*****
Clock cycle:          2
Instruction 20080001 is in ID stage
Fetched instruction 20120002
stallF:0
stallD:0
flushE:0
current pc: 00000004
*****
Clock cycle:          3
Instruction 20080001 is in ID stage
Instruction 20080001 is in EX stage
Fetched instruction 02484822
Forwarded 00000001 to writedataE from EX/MEM stage
current pc: 00000008
*****
Clock cycle:          4
Instruction 20120002 is in ID stage
Instruction 20080001 is in EX stage
Instruction 20080001 is in MEM stage
Fetched instruction 11120004
current pc: 0000000c
content of $t0 = 00000001
*****
Clock cycle:          5
Instruction 02484822 is in ID stage
Instruction 20120002 is in EX stage
Instruction 20080001 is in MEM stage
Instruction 20080001 is in WB stage
```

```
Fetched instruction 11090005
Forwarded 00000001 to writedataE from MEM/WB stage
Forwarded 00000002 to srcaE from EX/MEM stage
Branch not taken
current pc: 00000010
content of $t0 = 00000001
*****
Clock cycle: 6
Instruction 11120004 is in ID stage
Instruction 02484822 is in EX stage
Instruction 20120002 is in MEM stage
Instruction 20080001 is in WB stage
Fetched instruction 20100001
Branch is taken! Squashing instruction 20100001
current pc: 00000014
content of $s2 = 00000002
*****
Clock cycle: 7
Instruction 11120004 is in EX stage
Instruction 02484822 is in MEM stage
Instruction 20120002 is in WB stage
Fetched instruction ac120008
current pc: 00000028
content of $t1 = 00000001
*****
Clock cycle: 8
Instruction 11090005 is in EX stage
Instruction 11120004 is in MEM stage
Instruction 02484822 is in WB stage
Fetched instruction 0112882a
current pc: 0000002c
*****
```

```
Clock cycle:          9
Instruction ac120008 is in ID stage
Instruction 11090005 is in MEM stage
Instruction 11120004 is in WB stage
Fetched instruction 8c130008
current pc: 00000030
*****
Clock cycle:          10
Instruction 0112882a is in ID stage
Instruction ac120008 is in EX stage
Instruction 11090005 is in WB stage
Fetched instruction 0271a020
current pc: 00000034
no data written to register file
*****
Clock cycle:          11
address 00000008 now has data 00000002
Instruction 8c130008 is in ID stage
Instruction 0112882a is in EX stage
Instruction ac120008 is in MEM stage
lwstall=1
stallF:1
stallD:1
flushE:1
current pc: 00000038
*****
Clock cycle:          12
Instruction 0112882a is in MEM stage
Instruction ac120008 is in WB stage
stallF:0
stallD:0
flushE:0
```

```
current pc: 00000038
content of $s1 = 00000001
*****
Clock cycle: 13
Instruction 0271a020 is in ID stage
Instruction 8c130008 is in MEM stage
Instruction 0112882a is in WB stage
Forwarded 00000002 to srcaE from MEM/WB stage
stallF:x
stallD:x
flushE:x
current pc: 0000003c
content of $s3 = 00000002
*****
Clock cycle: 14
Instruction 0271a020 is in EX stage
Instruction 8c130008 is in WB stage
current pc: 00000040
*****
Clock cycle: 15
Instruction 0271a020 is in MEM stage
current pc: 00000044
content of $s4 = 00000003
*****
Clock cycle: 16
Instruction 0271a020 is in WB stage
current pc: 00000048
*****
Clock cycle: 17
current pc: 0000004c
*****
```

7. FPGA Results

To test the implementation on the FPGA we calculate the factorial of a number and display the value of the \$s1 register every clock cycle on the 6 seven-segment displays.

The clock is given as a push-button switch.

Code:

```
# fact_n => $s1
# count => $s0
# count => $t1
.text

INIT:
    addi $s0, $zero, 0x6
    addi $t0, $zero, 0x1
    addi $s1, $zero, 0x1

LOOP:
    beq $t0, $s0, DONE
    mul $s1, $s1, $t0
    addi $t0, $t0, 0x1
    j LOOP

DONE:
    addi $t9, $zero, 0x1
```

Output Log File:

```
current pc: xxxxxxxx
Fetched instruction 20100006
*****
Clock cycle:      1
current pc: 00000000
*****
Clock cycle:      2
```

```
Instruction 20100006 is in ID stage
Fetched instruction 20080001
stallF:0
stallD:0
flushE:0
current pc: 00000004
*****
Clock cycle:          3
Instruction 20100006 is in ID stage
Instruction 20100006 is in EX stage
Fetched instruction 20110001
Forwarded 00000006 to writedataE from EX/MEM stage
current pc: 00000008
*****
Clock cycle:          4
Instruction 20080001 is in ID stage
Instruction 20100006 is in EX stage
Instruction 20100006 is in MEM stage
Fetched instruction 11100004
current pc: 0000000c
content of $s0 = 00000006
*****
Clock cycle:          5
Instruction 20110001 is in ID stage
Instruction 20080001 is in EX stage
Instruction 20100006 is in MEM stage
Instruction 20100006 is in WB stage
Fetched instruction 72288802
Branch not taken
current pc: 00000010
content of $s0 = 00000006
*****
```

```

Clock cycle:      6
Instruction 11100004 is in ID stage
Instruction 20110001 is in EX stage
Instruction 20080001 is in MEM stage
Instruction 20100006 is in WB stage
Fetched instruction 21080001
Forwarded 00000001 to srcaE from MEM/WB stage
current pc: 00000014
content of $t0 = 00000001
*****
```

```

Clock cycle:      7
Instruction 72288802 is in ID stage
Instruction 11100004 is in EX stage
Instruction 20110001 is in MEM stage
Instruction 20080001 is in WB stage
Fetched instruction 08000003
current pc: 00000018
content of $s1 = 00000001
*****
```

```

Clock cycle:      8
Instruction 21080001 is in ID stage
Instruction 72288802 is in EX stage
Instruction 11100004 is in MEM stage
Instruction 20110001 is in WB stage
Fetched instruction 20190001
current pc: 0000001c
*****
```

```

Clock cycle:      9
Instruction 08000003 is in ID stage
Instruction 21080001 is in EX stage
Instruction 72288802 is in MEM stage
Instruction 11100004 is in WB stage

```

```
Fetched instruction 11100004
current pc: 0000000c
content of $s1 = 00000001
*****
Clock cycle:      10
Instruction 20190001 is in ID stage
Instruction 08000003 is in EX stage
Instruction 21080001 is in MEM stage
Instruction 72288802 is in WB stage
Fetched instruction 72288802
Branch not taken
current pc: 00000010
content of $t0 = 00000002
*****
Clock cycle:      11
Instruction 11100004 is in ID stage
Instruction 20190001 is in EX stage
Instruction 08000003 is in MEM stage
Instruction 21080001 is in WB stage
Fetched instruction 21080001
current pc: 00000014
*****
Clock cycle:      12
Instruction 72288802 is in ID stage
Instruction 11100004 is in EX stage
Instruction 20190001 is in MEM stage
Instruction 08000003 is in WB stage
Fetched instruction 08000003
current pc: 00000018
no data written to register file
*****
Clock cycle:      13
```

```
Instruction 21080001 is in ID stage
Instruction 72288802 is in EX stage
Instruction 11100004 is in MEM stage
Instruction 20190001 is in WB stage
Fetched instruction 20190001
current pc: 0000001c
*****
Clock cycle: 14
Instruction 08000003 is in ID stage
Instruction 21080001 is in EX stage
Instruction 72288802 is in MEM stage
Instruction 11100004 is in WB stage
Fetched instruction 11100004
current pc: 0000000c
content of $s1 = 00000002
*****
Clock cycle: 15
Instruction 20190001 is in ID stage
Instruction 08000003 is in EX stage
Instruction 21080001 is in MEM stage
Instruction 72288802 is in WB stage
Fetched instruction 72288802
Branch not taken
current pc: 00000010
content of $t0 = 00000003
*****
Clock cycle: 16
Instruction 11100004 is in ID stage
Instruction 20190001 is in EX stage
Instruction 08000003 is in MEM stage
Instruction 21080001 is in WB stage
Fetched instruction 21080001
```

```
current pc: 00000014
*****
Clock cycle:      17
Instruction 72288802 is in ID stage
Instruction 11100004 is in EX stage
Instruction 20190001 is in MEM stage
Instruction 08000003 is in WB stage
Fetched instruction 08000003
current pc: 00000018
no data written to register file
*****
Clock cycle:      18
Instruction 21080001 is in ID stage
Instruction 72288802 is in EX stage
Instruction 11100004 is in MEM stage
Instruction 20190001 is in WB stage
Fetched instruction 20190001
current pc: 0000001c
*****
Clock cycle:      19
Instruction 08000003 is in ID stage
Instruction 21080001 is in EX stage
Instruction 72288802 is in MEM stage
Instruction 11100004 is in WB stage
Fetched instruction 11100004
current pc: 0000000c
content of $s1 = 00000006
*****
Clock cycle:      20
Instruction 20190001 is in ID stage
Instruction 08000003 is in EX stage
Instruction 21080001 is in MEM stage
```

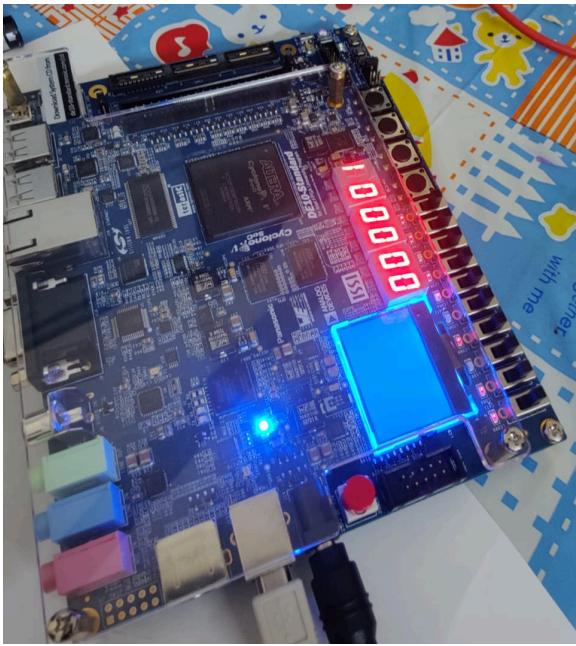
```
Instruction 72288802 is in WB stage
Fetched instruction 72288802
Branch not taken
current pc: 00000010
content of $t0 = 00000004
*****
Clock cycle: 21
Instruction 11100004 is in ID stage
Instruction 20190001 is in EX stage
Instruction 08000003 is in MEM stage
Instruction 21080001 is in WB stage
Fetched instruction 21080001
current pc: 00000014
*****
Clock cycle: 22
Instruction 72288802 is in ID stage
Instruction 11100004 is in EX stage
Instruction 20190001 is in MEM stage
Instruction 08000003 is in WB stage
Fetched instruction 08000003
current pc: 00000018
no data written to register file
*****
Clock cycle: 23
Instruction 21080001 is in ID stage
Instruction 72288802 is in EX stage
Instruction 11100004 is in MEM stage
Instruction 20190001 is in WB stage
Fetched instruction 20190001
current pc: 0000001c
*****
Clock cycle: 24
```

```
Instruction 08000003 is in ID stage
Instruction 21080001 is in EX stage
Instruction 72288802 is in MEM stage
Instruction 11100004 is in WB stage
Fetched instruction 11100004
current pc: 0000000c
content of $s1 = 00000018
*****
Clock cycle: 25
Instruction 20190001 is in ID stage
Instruction 08000003 is in EX stage
Instruction 21080001 is in MEM stage
Instruction 72288802 is in WB stage
Fetched instruction 72288802
Branch not taken
current pc: 00000010
content of $t0 = 00000005
*****
Clock cycle: 26
Instruction 11100004 is in ID stage
Instruction 20190001 is in EX stage
Instruction 08000003 is in MEM stage
Instruction 21080001 is in WB stage
Fetched instruction 21080001
current pc: 00000014
*****
Clock cycle: 27
Instruction 72288802 is in ID stage
Instruction 11100004 is in EX stage
Instruction 20190001 is in MEM stage
Instruction 08000003 is in WB stage
Fetched instruction 08000003
```

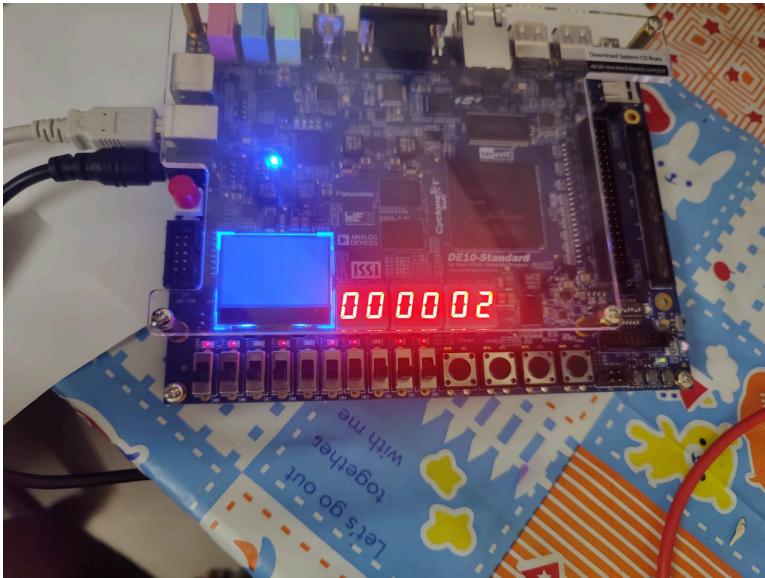
```
current pc: 00000018
no data written to register file
*****
Clock cycle:      28
Instruction 21080001 is in ID stage
Instruction 72288802 is in EX stage
Instruction 11100004 is in MEM stage
Instruction 20190001 is in WB stage
Fetched instruction 20190001
current pc: 0000001c
*****
Clock cycle:      29
Instruction 08000003 is in ID stage
Instruction 21080001 is in EX stage
Instruction 72288802 is in MEM stage
Instruction 11100004 is in WB stage
Fetched instruction 11100004
current pc: 0000000c
content of $s1 = 00000078
*****
Clock cycle:      30
Instruction 20190001 is in ID stage
Instruction 08000003 is in EX stage
Instruction 21080001 is in MEM stage
Instruction 72288802 is in WB stage
Fetched instruction 72288802
Branch not taken
current pc: 00000010
content of $t0 = 00000006
Branch is taken! Squashing instruction 72288802
*****
Clock cycle:      31
```

```
Instruction 20190001 is in EX stage
Instruction 08000003 is in MEM stage
Instruction 21080001 is in WB stage
current pc: 00000020
*****
Clock cycle:      32
Instruction 11100004 is in EX stage
Instruction 20190001 is in MEM stage
Instruction 08000003 is in WB stage
stallF:x
stallD:x
flushE:x
current pc: 00000024
no data written to register file
*****
Clock cycle:      33
Instruction 11100004 is in MEM stage
Instruction 20190001 is in WB stage
current pc: 00000028
*****
Clock cycle:      34
Instruction 11100004 is in WB stage
current pc: 0000002c
no data written to register file
*****
Clock cycle:      35
current pc: 00000030
*****
```

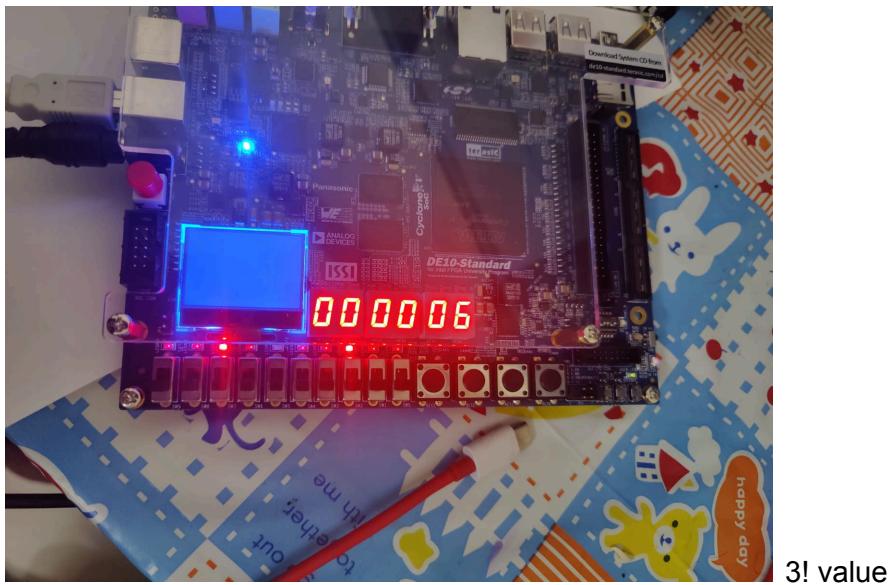
FPGA output:



The initial value of \$s1 register with the value 1 (1!)



2! value



Resource Utilization:

Project Navigator Entity/Instance

Cyclone V: 5CSXFC6D6F31C6
M_MIPS_CPU

Tasks Compilation

- Task
 - Analysis & Elaboration
 - Partition Merge
 - Netlist Viewers
 - Design Assistant (Post-Mapping)
 - I/O Assignment Analysis
 - Fitter (Place & Route) 00:01
 - Assembler (Generate programming files) 00:00
 - Timing Analysis 00:00
 - EDA Netlist Writer
 - Edit Settings
 - Program Device (Open Programmer)

Compilation Report - MIPS-5-Stage-Pipeline-FPGA x M_ADDER.v x

Table of Contents

Flow Summary

Flow Status	Successful - Sun Mar 17 22:58:11 2024
Quartus Prime Version	22.1std.1 Build 917 02/14/2023 SC Lite Edition
Revision Name	MIPS-5-Stage-Pipeline-FPGA
Top-level Entity Name	M_MIPS_CPU
Family	Cyclone V
Device	5CSXFC6D6F31C6
Timing Models	Final
Logic utilization (in ALMs)	255 / 41,910 (< 1 %)
Total registers	269
Total pins	109 / 499 (22 %)
Total virtual pins	0
Total block memory bits	2,048 / 5,662,720 (< 1 %)
Total DSP Blocks	2 / 112 (2 %)
Total HSSI RX PCSS	0 / 9 (0 %)
Total HSSI PMA RX Deserializers	0 / 9 (0 %)
Total HSSI TX PCSS	0 / 9 (0 %)
Total HSSI PMA TX Serializers	0 / 9 (0 %)
Total PLLs	0 / 15 (0 %)
Total DLLs	0 / 4 (0 %)

Messages

Type	ID	Message
332146	worst_case_minimum_pulse_width_slack	worst_case minimum pulse width slack is -2.174.
332102	Design is not fully constrained for setup requirements	Design is not fully constrained for hold requirements
332102	Quartus Prime Timing Analyzer was successful.	Quartus Prime Timing Analyzer was successful. 0 errors, 6 warnings
18236	Running Quartus Prime EDA Netlist Writer	*****
18236	Command: quartus_edt --read_settings_files=off --write_settings_files=off MIPS-5-Stage-Pipeline-FPGA -c MIPS-5-Stage-Pipeline-FPGA	Command: quartus_edt --read_settings_files=off --write_settings_files=off MIPS-5-Stage-Pipeline-FPGA -c MIPS-5-Stage-Pipeline-FPGA
204019	Number of processors has not been specified which may cause overloading on shared machines. Set the global assignment NUM_PARALLEL_PROCESSORS to	Number of processors has not been specified which may cause overloading on shared machines. Set the global assignment NUM_PARALLEL_PROCESSORS to
204019	Generated file MIPS-5-Stage-Pipeline-FPGA.vo in folder "C:/Users/lnba2/Desktop/lab/MIPS-5-Stage-Pipeline/MIPS-5-Stage-Pipeline-FPGA/simul"	Generated file MIPS-5-Stage-Pipeline-FPGA.vo in folder "C:/Users/lnba2/Desktop/lab/MIPS-5-Stage-Pipeline/MIPS-5-Stage-Pipeline-FPGA/simul"

RTL netlist:

