

## Tutorial - 10

### Problem-I

Filter out the instructions that may lead to control hazards and data hazards in a five stage pipelined processor following RISC-V ISA.

Instructions					
JAL	LW	ADD	SW	BEQ	SUB

### Problem-II

Consider two different machines. The first has a single-stage, non-pipelined machine with a cycle time of 15 ns. The second is a pipelined machine with 5 pipeline stages and a cycle time of 3ns.

- What is the speedup of the pipelined machine versus the single-cycle machine, assuming there are no stalls?
- What is the speedup of the pipelined machine versus the single cycle machine if the pipeline stalls 1 cycle for 25% of the instructions?
- Now consider a 4-stage pipeline machine with a cycle time of 3.1 ns. Again, assuming no stalls, is this implementation faster or slower than the original 5-stage pipeline? Explain your answer.

### Problem-III

You need to reorganize or optimize the below-mentioned assembly program such that the functionality of the program remains unchanged and the total execution time is minimum. Assume the processor chosen for the execution is 5-stage and follows RISC-V ISA and has full data forwarding capability.

#### Assembly Program

```
0x00: Addi x4,x0,#4
0x04: Sub x5,x3,x4
0x08: Lw x6,8(x5)
0x0C: Add x7,x6,x0
0x10: Lw x8,12(x5)
0x14: Add x9,x8,x0
0x18: Addi x3,x0,#3
0x1C: Add x9,x9,x3
```

### Problem-IV

Execute the below mentioned assembly program and explain the use of stack memory in this program.

#### Assembly Program

```
Addi x4,x0,#4
Addi x3,x1,x0
Add x5,x3,x1
Sub sp,sp,x4
Sw x3,0(sp)
Add x3,x5,x5
Add x5,x5,x0
```

```
Lw x3,0(sp)
Add sp,sp,s4
Beq x0,x0,#0
```

### **Problem-V**

Execute the below-mentioned assembly program on a processor having standard five-stages and following RISC-V ISA. And find the difference in total number of execution cycles among the two cases

- (a) All branches are predicted correctly
- (b) All branches are predicted incorrectly

### Assembly Program

```
Addi x1,x0,#1
Addi x10,x0,#10
```

### **DO\_AGAIN:**

```
Sub x10,x10,x1
Beq x10, x0, DO_AGAIN
Beq x0,x0,#0
```