

# Explanation of Key Sections in the Code

To ensure that the reader can understand and modify the TestBench if needed, let's go through the main parts of the code:

## 1. Instantiating Flip-Flops

```
D_FLIP_FLOP FF1 (.D(b0), .sync_reset(Reset), .Q(Q1));
```

```
D_FLIP_FLOP FF2 (.D(Q1), .sync_reset(Reset), .Q(Q2));
```

```
D_FLIP_FLOP FF3 (.D(Q2), .sync_reset(Reset), .Q(led0));
```

- Three Flip-Flops are instantiated and connected in cascade.
- The first Flip-Flop receives the input signal b0, and each subsequent Flip-Flop takes the Q output of the previous one as input.
- The last output (led0) represents the divided frequency after three Flip-Flop stages.

## 2. Clock Signal Generation

```
always #5 b0 = ~b0; // Toggle b0 every 5 time units (creates clock signal)
```

- This command toggles b0 every 5 time units, effectively simulating a 10-time unit period clock.
- As a result, b0 acts as a square wave signal.

## 3. Reset Initialization

```
initial begin
```

```
Reset = 1;
```

```
b0 = 0;
```

```
#10 Reset = 0; // Release reset
```

***end***

- The Reset signal is initially set to 1 to clear all Flip-Flop outputs.
- After 10 time units, Reset is set to 0, allowing normal operation of the Flip-Flops.

#### **4. Simulation Control**

***initial begin***

***\$dumpfile("testbench.vcd");***

***\$dumpvars(0, testbench);***

***#1000 \$finish; // Run the simulation for sufficient time***

***end***

- The \$dumpfile("testbench.vcd") command stores all simulation data in a .vcd file for waveform analysis.
- The simulation runs for 1000 time units, ensuring enough time to observe multiple frequency divisions.

---

Revision #1

Created 18 February 2025 19:29:30 by Caroline

Updated 18 February 2025 19:31:50 by Caroline