

Verilog HDL: Timing and Delays

Pravin Zode

Outline

- ``timescale`
- `$timeformat`
- `$printtimescale`
- Measure time

timescale

- Defines time units and precision for simulation
- Syntax: `timescale <time_unit> / <time_precision>`
- Example: `timescale 1ns / 1ps`
 - Time unit: 1 nanosecond.
 - Time precision: 1 picosecond
- Affects simulation timing and delay calculations.

Example: timescale

```
1  `timescale 1ns/1ps
2  module timescale_example;
3      initial begin
4          #10 $display("Time: %t", $time);
5      end
6  endmodule
7
8  module tb_timescale_example;
9      timescale_example uut();
10     initial #20 $finish;
11 endmodule
```

Output:

```
Time: 10
```

\$timeformat

- Controls how simulation time is displayed in output
- Syntax:
 - `$timeformat(unit, precision, suffix, min_field_width);`
- Parameters:
 - `unit`: Base time unit (e.g., 0 for seconds, -9 for nanoseconds)
 - `precision`: Number of decimal places
 - `suffix`: Character string to display after time (e.g., "ns")
 - `min_field_width`: Minimum width of displayed time value
- Example: `$timeformat(-9, 2, " ns", 10);`

Example: \$timeformat

```
1  module timeformat_example;
2      initial begin
3          $timeformat(-9, 2, " ns", 10);
4          #5 $display("Formatted Time: %t", $time);
5      end
6  endmodule
7
8  module tb_timeformat_example;
9      timeformat_example uut();
10     initial #10 $finish;
11 endmodule
```

Output:

```
Formatted Time: 5 ns
```

\$prnttimescale

- Displays the current timescale settings of a module
- Syntax: \$prnttimescale(<module_name>)

```
1  √ module prnttimescale_example;  
2    |   initial $prnttimescale(prnttimescale_example);  
3    | endmodule  
4  
5  √ module tb_prnttimescale_example;  
6    |   prnttimescale_example uut();  
7    |   initial #5 $finish;  
8    | endmodule
```

Output:

```
Time scale of (prnttimescale_example) is 1 ns / 1 ps
```

Comparison Between `$time`, `$stime`, and `$realtime`

Function	Data Type	Precision	Use Case
<code>\$time</code>	Integer	Low	Coarse time measurement
<code>\$stime</code>	32-bit Integer	Medium	Medium precision timing
<code>\$realtime</code>	Real Number	High	High-precision time measurement

Comparison Between \$time, \$stime, and \$realtime

```
1  module time_comparison;
2      initial begin
3          $display("$time: %t", $time);
4          $display("$stime: %0d", $stime);
5          $display("$realtime: %0.3f", $realtime);
6      end
7  endmodule
8
9  module tb_time_comparison;
10     time_comparison uut();
11     initial #10 $finish;
12 endmodule
```

Output:

```
$time: 0
$stime: 0
$realtime: 0.000
```

Example 1 - Using \$time and \$realtime in a Counter

```
1  module counter_example;
2      reg clk;
3      integer count = 0;
4      initial begin
5          clk = 0;
6          forever #5 clk = ~clk;
7      end
8      always @(posedge clk) begin
9          count = count + 1;
10         $display("Time: %d ns, Count: %d", $time, count);
11     end
12 endmodule
13
14 module tb_counter_example;
15     counter_example uut();
16     initial #50 $finish;
17 endmodule
```

Output:

```
Time: 5 ns, Count: 1
Time: 15 ns, Count: 2
Time: 25 ns, Count: 3
...
```

Example 2 - Measuring Delay Using \$realtime

```
1 module delay_measure;
2     reg signal;
3     real start_time, end_time, delay;
4     initial begin
5         signal = 0;
6         #7 signal = 1;
7         start_time = $realtime;
8         #12 signal = 0;
9         end_time = $realtime;
10        delay = end_time - start_time;
11        $display("Measured delay: %0.3f ns", delay);
12    end
13 endmodule

14
15 module tb_delay_measure;
16     delay_measure uut();
17     initial #30 $finish;
18 endmodule
```

Output:

```
Measured delay: 12.000 ns
```

Example 3 – Time Test

```
1  module time_test;
2      initial begin
3          $timeformat(-9, 2, " ns", 10);
4          #5 $display("Current Simulation Time: %t", $time);
5          #10 $display("Current Simulation Time: %t", $time);
6      end
7  endmodule
8
9  module tb_time_test;
10     time_test uut();
11     initial #20 $finish;
12 endmodule
```

Output:

```
Current Simulation Time: 5 ns
Current Simulation Time: 15 ns
```

Summary

- Accurate timing is essential for simulation and verification of digital designs.
- ``timescale` sets the resolution and precision of simulation delays.
- `$timeformat` customizes the format of time values in output displays.
- `$printtimescale` helps verify module timescale settings during simulation.
- `$time`, `$realtime`, and `$stime` allow designers to measure and track simulation time, useful in debugging and testbench analysis.



Thank you !

Happy Learning