# DAY-62
# #100DAYSRTL

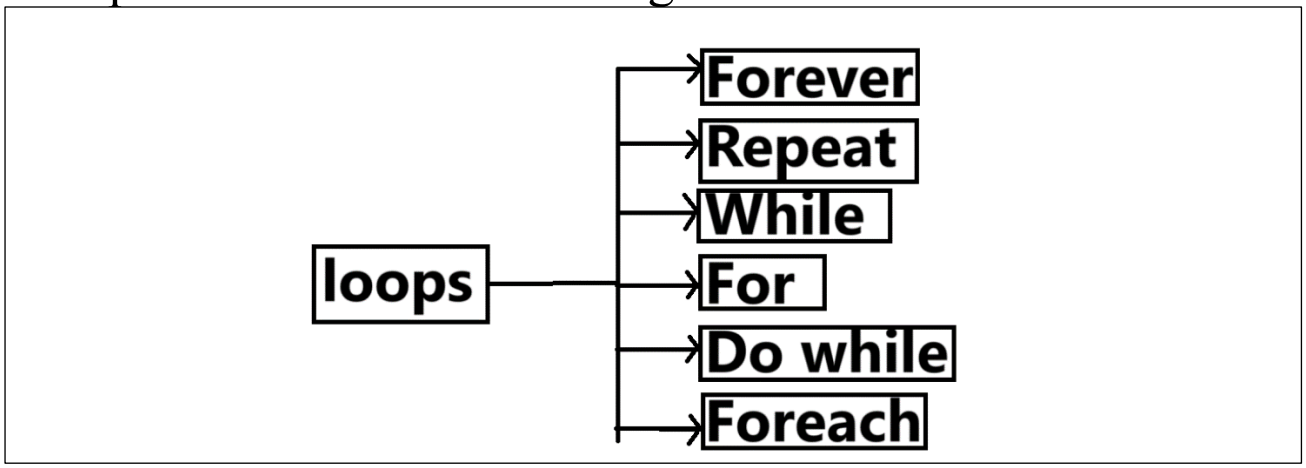
## "System Verilog: Loops"

## "Introduction":-

Loops are statements that help to perform repetitive tasks by writing them once. In testbench, loops are very important as we must perform lots of repetitive tasks such as sampling a signal after a certain duration. So, this sampling should happen repeatedly until the simulation ends.

## "Types Of loops":-

1. **Finite loop** – this type of loop will repeat for certain number until some conditions are not met. Once the condition evaluates to be false it will move out of the loop and execute the statements present after loop.
2. **Infinite loop** – this type of loop basically run for infinite iterations or until the simulation ends. In this type of loops either there is no condition evaluation, or the condition evaluates to true every time. Infinite loops can be exited with the help of break statement if required. Infinite loops should be used with great care because using it in a wrong way can lead to unexpected results or also hang the simulation.



# “Forever”:-

This is an infinite loop, just like while (True) .
**Note** that your simulation will hang unless you include a time delay inside the forever block to advance simulation time

**Code practicing:-**

```
module tb;
  initial begin
    forever begin
      #5;
      $display("Semicon");
    end
  end
  initial begin
    #50;$finish();
  end
endmodule
```

**Result:-**

```
Semicon
Semicon
Semicon
Semicon
Semicon
Semicon
Semicon
Semicon
Semicon
Simulation complete via $finish(1) at time 50 NS + 0
```

# “Repeat”:-

Used to repeat statements in a block a certain number of times.

**Code practicing:-**

```
module tb;
  initial begin
    repeat(5) begin
      #5;
      $display("Semicon");
    end
  end
endmodule
```

**Result:-**

```
Semicon
Semicon
Semicon
Semicon
Semicon
xmsim: *W,RNQUIE: Simulation is complete.
```

# “While”:-

It'll repeat the block as long as the condition is true

## Code practicing:-

```
module tb;
  int i=0;
  initial begin
    while (i<10) begin
      $display("i=%d;",i);
      i=i+1;#10;
  end
  end
endmodule
```

## Result:-

```
i=         0;
i=         1;
i=         2;
i=         3;
i=         4;
i=         5;
i=         6;
i=         7;
i=         8;
i=         9;
```

# “For loop”:-

It'll repeat the block as long as the condition is true

## Code practicing:-

```
module tb;
  initial begin
    for (int i=0;i<10;i=i+1) begin
      $display("i=%d;",i);
      #10;
  end
  end
endmodule
```

## Result:-

```
i=         0;
i=         1;
i=         2;
i=         3;
i=         4;
i=         5;
i=         6;
i=         7;
i=         8;
i=         9;
```

# “Do While”:-

Repeats the given set of statements at least once, and then loops as long as the condition is true

## Code practicing:-

```
module tb;
  int a=1;
  initial begin
    do begin
      $display("Hi");#5;
    end while(a>10) ;
  end
endmodule
```

## Result:-

```
Hi
```

# “Foreach”:-

Used mainly to iterate through all elements in an array

## Code practicing:-

```
module tb;
  int a[3]='{1,2,3};
  initial begin
    foreach(a[i]) begin
      $display("a[%d]=%d",i,a[i]);
    end
  end
endmodule
```

## Result:-

```
a[          0]=          1
a[          1]=          2
a[          2]=          3
```