Experiment 5 – Tasks and Functions

PART 1:

Aim:

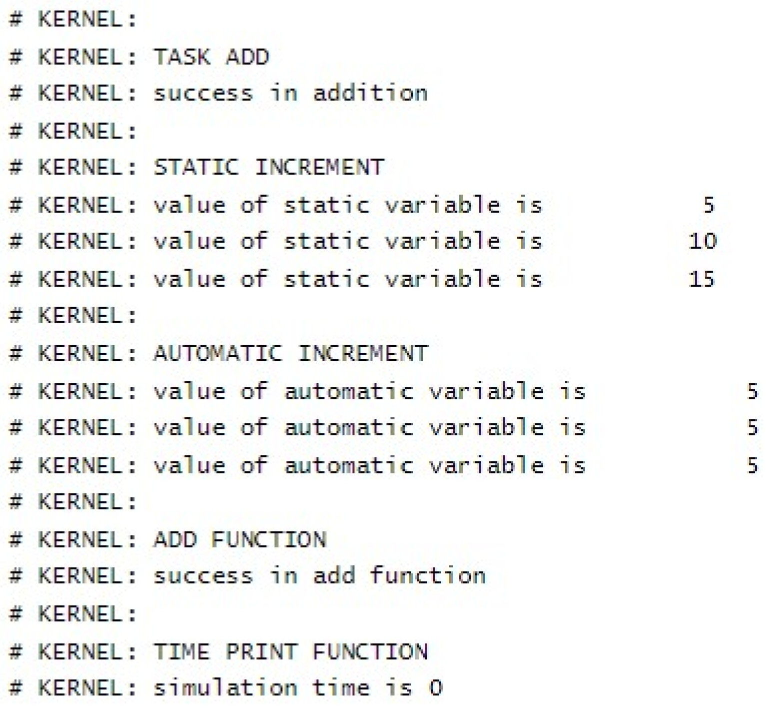
- Write the SV “task” code for the addition of two integer numbers and check it correctness.
- Observe the difference between simple and automatic tasks:
    - Write a simple task which increments the value of a local variable by a specified amount.
    - Write an automatic task which increments the value of a local variable by a specified amount.
- Write the code using SV “function” for the addition of two integer numbers.
- Write the SV void function to print the current simulation time. Check its corrections.

Code:

```
module add();
  int y;

  task sum(input int a,b, output int y);
    y = a+b;
  endtask

  task stat_inc;
    begin
      static int g = 0;
      g += 5;
      $display("value of static variable is %d",g);
    end
  endtask

  task auto_inc;
    begin
      automatic int u = 0;
      u += 5;
      $display("value of automatic variable is %d",u);
    end
  endtask

  function int add(int x, int y);
    add = x+y;
  endfunction

  function void time_print();
```

```
    $display("simulation time is %0d",$time);
  endfunction

  initial begin
    int a = $random;
    int b = $random;

    $display("\n\nTASK ADD");
    sum(a,b,y);
    if(y == a+b) $display("success in addition");
    else $display("failure");

    $display("\nSTATIC INCREMENT");
    repeat(3) stat_inc();
    $display("\nAUTOMATIC INCREMENT");
    repeat(3) auto_inc();

    $display("\nADD FUNCTION");
    y= add(a,b);
    if(y == a+b) $display("success in add function");
    else $display("failure");

    $display("\nTIME PRINT FUNCTION");
    time_print();
    $display("\n\n");

  end
endmodule
```

Output:



```
# KERNEL:
# KERNEL: TASK ADD
# KERNEL: success in addition
# KERNEL:
# KERNEL: STATIC INCREMENT
# KERNEL: value of static variable is           5
# KERNEL: value of static variable is          10
# KERNEL: value of static variable is          15
# KERNEL:
# KERNEL: AUTOMATIC INCREMENT
# KERNEL: value of automatic variable is           5
# KERNEL: value of automatic variable is           5
# KERNEL: value of automatic variable is           5
# KERNEL:
# KERNEL: ADD FUNCTION
# KERNEL: success in add function
# KERNEL:
# KERNEL: TIME PRINT FUNCTION
# KERNEL: simulation time is 0
```

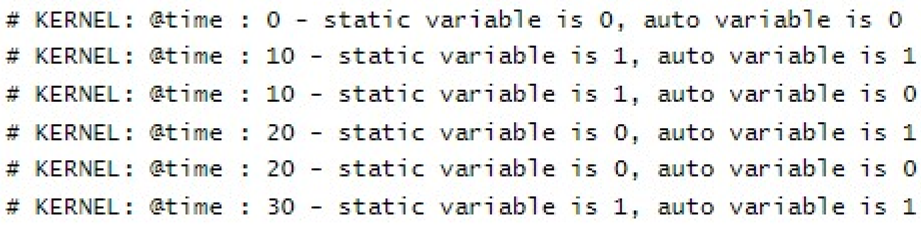
PART 2:

Aim: Demonstrate the difference between static and automatic variables used in SV task.

Code:

```
module auto_variable_task;

  task auto_delay(input time delay);
    logic static_var = 1'b0;
    automatic logic auto_var = 1'b0;

     $display("@time : %0d - static variable is %b, auto variable is
    %b",$time,static_var, auto_var);
    #delay;

    static_var = !static_var;
    auto_var = !auto_var;
    $display("@time : %0d - static variable is %b, auto variable is
    %b",$time,static_var, auto_var);

  endtask : auto_delay

  initial repeat(3) auto_delay(10ns);
endmodule
```

Output:



```
# KERNEL: @time : 0 - static variable is 0, auto variable is 0
# KERNEL: @time : 10 - static variable is 1, auto variable is 1
# KERNEL: @time : 10 - static variable is 1, auto variable is 0
# KERNEL: @time : 20 - static variable is 0, auto variable is 1
# KERNEL: @time : 20 - static variable is 0, auto variable is 0
# KERNEL: @time : 30 - static variable is 1, auto variable is 1
```

Result:

The given problem statement is executed and verified to be correct.