SV Assignment – 6

1. Create 3 parllel threads:
 Thread 1: takes 10 posedges to complete
 Thread 2: takes 5 posedges to complete
 Thread 3: takes 30 posedges to complete
Write a code to achieve above threads.

2. For the above program, after 5 posedges, simulation should be finished. Write a program to achieve it.

3. Write a program containing the following two threads.
T1: forever
```
        begin
        #1;
        addr=$random_range(10,100);
        end
```

T2: forever
```
        begin
        #2;
        $display(addr);
        end
```
a. After 20 time units T1, T2 should be stopped.
b. Display the addr values. Then finish the simulation

4. Write a program with the below two tasks.

```
int addr_q[$];
int data_q[$];

task drive(input int addr, input int data);
repeat(10)
begin
        #10;
        addr_q.push_front(addr);
        data_q.push_front(data);
end
endtask

task display();
repeat(10)
begin
        #20;
        $display(“addr queue =%p”,addr_q);
$display(“data queue =%p”,data_q);
end
endatask
```

a. Call these two tasks parallely. As soon as the this parallel process gets started, it should come out and execute the next lines of the code.

b. The next lines of code include: Checking whether queue is empty or not. If queue is empty, then hold untill the above threads gets completed. If queue is not empty, the display the contents of the queue.

5.
a. Write code for memory model.(RTL)
b.Introduce Generator, Transaction classes.
c. Also have driver, input monitor, output monitor and scoreboard.

6. Write a program with two classes(Generator, Driver).
Generator – Generate random values to an array and give this to driver.
Driver – Take the array and display the content of an array(queue type).
Generator has to send an array to the driver for four times.
a. Achieve this without using any inbuilt concepts.
b. Achieve this using inbuilt cconcepts.

7. Write a program with three classes(Transaction, Generator, Driver).
Transaction Class ---> addr – 8 bit
data – 16 bit
Generator – Generate the transaction for 4 times
Driver – After the 4 transactions are generated, then only the driver has to take those transactions and display them.
a. Driver should take 4 transactions without deleting the content in the mailbox. Also display the mailbox contents and its size.
b. Driver should take 4 transactions by erasing the content in mailbox. (once content get accessed then that content is erased)

8.Write a program with three classes(Transaction, Generator, Driver).
Transaction Class ---> addr – 8 bit
data – 16 bit
Generator – Generate one transaction
Driver – After the generation of one transaction, the driver has to take that transaction and display it. Repeat the above generation and driving of transaction for four times.
a. Driver should take 4 transactions without deleting the content in the mailbox. Also display the mailbox contents and its size.
b. Driver should take 4 transactions by erasing the content in mailbox. (once content get accessed then that content is erased)

9.Write a program with three classes(Transaction, Generator, Driver).
Transaction class --> int arr_que[$]; (Write a constraint for queue size.)
Generator – For every 10 time units generate a transaction. Repeat for 2 times.
Driver – In driver the valid data will be available only after 10 timeunits (i.e.which wiil be given by the generator),so before 10 times units itself the array value to be initialized with some default values in the driver.And after 10 timeunits take the content which is generated by the Generator.

10.Write a program with three classes(Generator, Driver, Monitor).
In these three classes, declare three different events and trigger them at different time units. Once the events are detected in sequence manner then end the simulation.

11.
Check the functionality of a counter by writing the self check testbench(means write a automation code to verify the counter) for

a. Counter - mod4

Ex:- If count values are in sequence 0,1,3,0 automation logic should give an error .If counter values are in sequence 0,1,2,3 then only pass. Other than this sequence of any others it should give an error. This we should check until the end of simulation.

b. counter - mod1000

Note:

Without RTL: use $readmemb/$readmemh to generate the counter values (assume that this is the design o/p)----easy approach