14/08/2023 - Questions

1. write a dflipflop code with asynchronous active low reset in verilog

```
//Design code
module dff(d,clk,reset,q,qbar);
  input d,clk,reset;
  output reg q;
  output qbar;
  assign qbar = ~q;
  always @(posedge clk or negedge reset)
    begin
      if(!reset)
        q = 0;
      else
        q = d;
    end
endmodule

//Verilog testbench
module tb;
  reg d, clk = 0, reset;
  wire q, qbar;
  dff uut(.d(d), .clk(clk), .reset(reset), .q(q), .qbar(qbar));
  always #5 clk = ~clk;
  initial begin
    $monitor("value of q at time %0d is %0d", $time, q);
    #2 d = 1;
    #10 d = 0;
    $dumpfile("dump.vcd"); $dumpvars;
    #50 $finish;
  end
endmodule

//Output
# value of q at time 0 is x
# value of q at time 5 is 1
# value of q at time 15 is 0
```

---

2. (a) Constraint for the array[100] to generate the values in the range of 1 to 50.
(b) For the above question, I want values 10 and 20 in between

```
//By using weighted distribution for 10 and 20 to appear multiple times
class pack;
rand int array[100];
  constraint c1{foreach(array[i])
    array[i] dist { [1:9]:/1 , 10:= 2,[11:19]:/1, 20:= 2, [21:50]:/1 };}
endclass
module top;
```

```
 pack pk = new();
 initial begin
   repeat(4)begin
     void'(pk.randomize());
   $display(" array = %p", pk.array);
   end
 end
endmodule

//By using constraint overriding method to display only 10 and 20
class pack;
rand int array[100];
 constraint c1{foreach(array[i])

   array[i] inside { [1:50] };}
endclass
module top;

 pack pk = new();
 initial begin
   repeat(4)begin
     void'(pk.randomize() with { foreach(array[i]) array[i] inside {10,20};});
   $display(" array = %p", pk.array);
   end
 end
endmodule
```

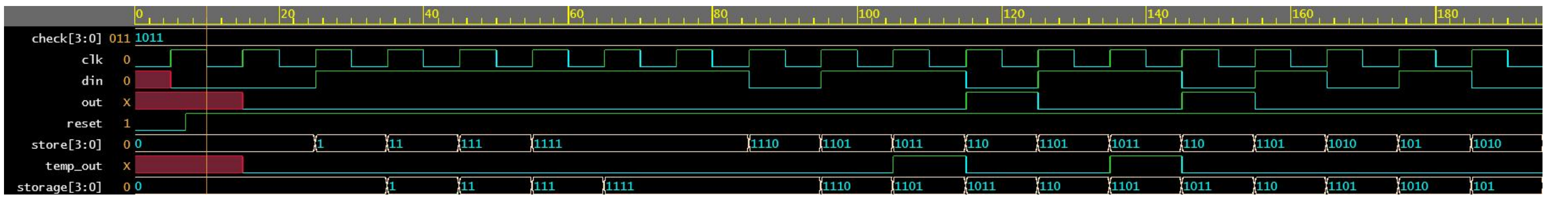
3. Construct dynamic array with 10 elements initially. Now I want to increase its size by '1' without loss of previous array

```
module top;
 int array[];
 initial begin
   array = new[10];
   array ='{ 1,2,3,4,5,6,7,8,9,10};
   array = new[array.size + 1](array);
   array[10] = 5;
   $display(" array = %p", array);

 end
endmodule

//Output
#  array = '{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 5}
```

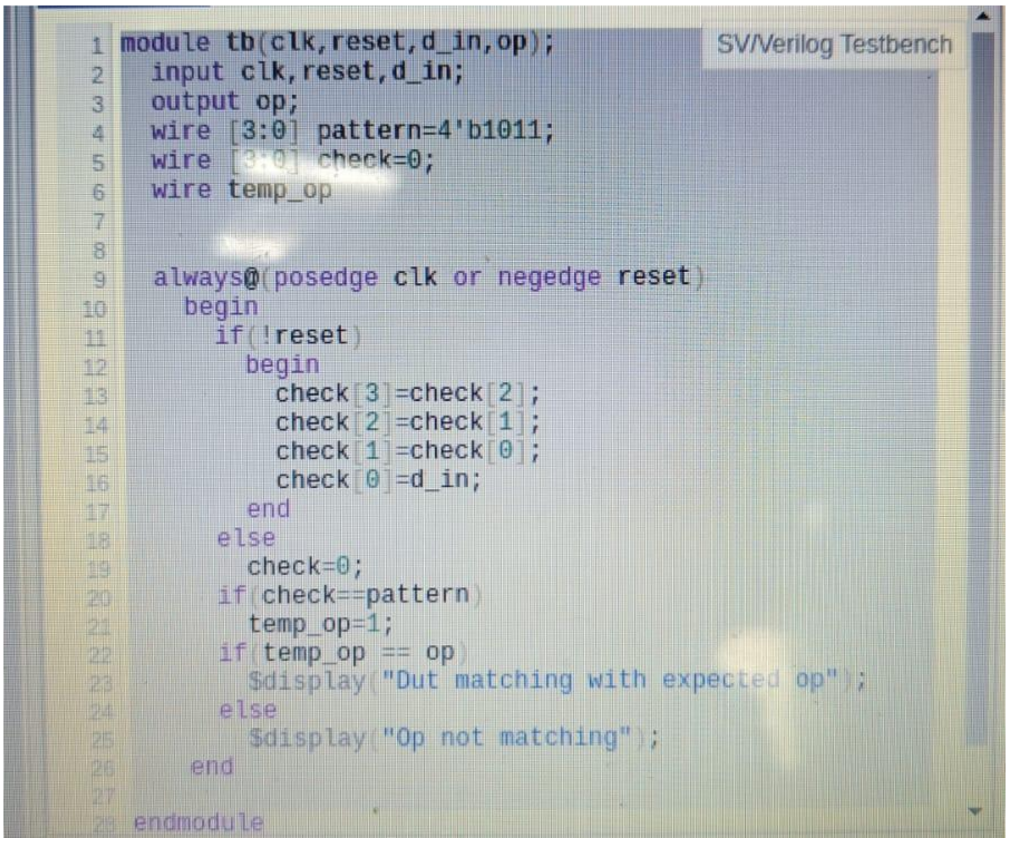
4. How to write a checker comparison in testbench(verilog) for state machine encoded pattern 1011?

```
//testbench
module seq_detector_tb;
  reg din,clk = 0,temp_out,reset;
  wire out;
  reg [3:0] check = 4'b1011;
  reg [3:0] store = 0;
  dut inst(.din(din), .clk(clk), .reset(reset), .out(out));
  always #5 clk = ~clk;
  initial begin
    reset = 0;
    #7 reset = 1;
  end
  always @(posedge clk or negedge reset)
    begin
      if(reset) begin
      din = $random();
      store = {store,din}; //can also use shift registers
      if(check == store)
        temp_out = 1;
      else
        temp_out = 0;
      if(temp_out == out)
        $display("Sequence detector is successfully detecting the sequence
@ %0d",$time);
      else
        $display("Error in detecting @ %0d", $time);
    end
      else
        din = 0;
    end
  initial begin
    $dumpfile("dump.vcd"); $dumpvars;
    #200 $finish;
  end
endmodule

//example design code
module dut(din,clk,reset,out);
  input din,clk,reset;
  output reg out;
  reg [3:0]storage = 0;
  always @(posedge clk)
    begin
      if(reset) begin
      storage = {storage,din};
      if(storage == 4'b1011)
         out = 1;
      else
        out = 0;
         end
    end
         endmodule
```

//output



//Akshay's code:



```
module tb(clk,reset,d_in,op);
  input clk,reset,d_in;
  output op;
  wire [3:0] pattern=4'b1011;
  wire [3:0] check=0;
  wire temp_op


  always@(posedge clk or negedge reset)
    begin
      if(!reset)
        begin
          check[3]=check[2];
          check[2]=check[1];
          check[1]=check[0];
          check[0]=d_in;
        end
      else
        check=0;
      if(check==pattern)
        temp_op=1;
      if(temp_op == op)
        $display("Dut matching with expected op");
      else
        $display("Op not matching");
     end

endmodule
```

5. Any difference between Ignore and Illegal bins

Illegal ->error/ fatal error(depends on simulator tool) when value is encountered
Ignore -> no msg when value is encountered
Both do not consider the bin value for the coverage percent.

6. What are pass by value and pass by reference with own example

```
module top;
  int a =1;
  int  b= 2;
```

```
 int c,d;
 initial begin
  pass_value(a,b);
   $display(" value: a = %0d, b= %0d", a,b);

   d= pass_ref(a,b);
   $display(" refer: a = %0d, b= %0d", a,b);
 end
 function int pass_value( input int a,b);
   int temp;
   temp =a;
   a= b;
   b=temp;
   $display("  inside pass_value a = %0d, b= %0d", a,b);
   endfunction

   function automatic int pass_ref(ref  int a, b);
     int temp1;
     temp1= a;
     a=b;
     b=temp1;
     $display(" inside refer: a = %0d, b= %0d", a,b);
   endfunction
endmodule
```

---

7. write a sequence which generates output has 1010.... we need to write constaint for this sequence and variable is 15 bit

```
class test;
 rand bit [15:0] a;

 constraint c{ foreach(a[i])
   if(i%2==0)
     a[i] inside {0};
         else
           a[i] inside {1};
         }
endclass

module top;
 test t;
 initial begin
   t=new();
   t.randomize();
   $display("%b",t.a);

 end
endmodule
```

‘OR’

```
class pack;
  randc bit [14:0] a;
  constraint c1{foreach(a[i])
    if(a[0] == 0)
      a[0] ==1;}
  constraint c2{ foreach(a[i])
    if(i>0)
      a[i] == ~a[i-1];}
endclass

module top;
  pack pk =new();
  initial begin
    repeat(4)begin
    void'(pk.randomize());
        $display(" array = %b", pk.a);
    end
  end
endmodule
```

---

8. there is array which is randomised and it has 1000 elements and one element is not randomised how you will find it is which element and what location.