1. What’s the output of the following code ?

class A;

  virtual function void display();

  $display("base class");

 endfunction

endclass

class B extends A;

   virtual function void display();

  $display( "B");

endfunction

endclass

class C extends B;

 endclass

class D extends C;

 virtual function void display();

  $display( "D");

endfunction

endclass

module poly;

initial  begin

A a=new();

B b= new();

C c= new();

D d=new();

$cast (a,b);

$cast (b,c);

$cast (c,d);

$cast (d,b);

a.display();

b.display();

c.display();

d.display();

end

endmodule

1. The below code waits for any of the event(A/B/C) before calling calc offset. Update the below code to wait for any of the 2 events to complete before calling calc\_offset function.

              fork

                 begin

                    @( eventA);

                 end

                begin

                    @( eventB);

                 end

                 begin

                    @( eventC);

                 end

             join\_any

            disable fork;

            calc\_offset();

1. Draw the FSM for an overlapping sequence detector 10110 and mention which FSM mealy or moore you have used.
2. Draw the w/f and hardware logic for the following set of codes. Assume initial values of b and d as 2 and 3

a.      always @ (posedge clk)

 begin

a = b;

c = d;

end

b.      always @ (posedge clk)

begin

a <= b;

c <= d;

end

1. write an Assertion for following conditions: -

a.      after request, ack should be high within 2 to 10 clock cycles

b.      Given a fifo depth 4, when fifo\_write is equals to fifo\_read, a flag is asserted

1. How you find in which location index value 100 is stored in an array?
2. Write a constraint for 32 bit address in which the number of ones is 10 and no ones is consecutive i.e two bits next to each other are not 1 ?
3. There is a 16-bit arbiter. At any time, one bit is enabled to activate the corresponding master. Write a constraint to implement this
4. What is the output of this snippet?

 Initial

            begin

               for (int i = 0 ; i< 10; i++)

                 begin

                         $display(value of i = %0d”, i);

                 end

 for (int i = 0 ; i< 10; i++)

               begin

                  $monitor(value of i = %0d”, i);

                end

        end

1. Explain the concept of Polymorphism
2. One full pizza you need to cut the pieces of 64, how you do and how many steps you will take to achieve the 64 pieces?
3. Write a sv code for Fibonacci series?
4. What is the output of the following code?

**module test;**

**bit [31:0] abc[\*];**

**initial begin**

**abc[500] = 40;**

**$display("size of abc %0d, ",abc.num());**

**end**

**endmodule**

A: size of abc 500

B: size of abc 501

C: size of abc 40

D: size of abc 1

1. Consider a Dynamic array having 0 to 10, need to get display greater than 5?
2. Write 8\*1 mux with 2\*1 mux and then I will swap the selection points (s2,s1,s0 =>s0,s1,s2) have to get the same output. what would you change?
3. What is the output of the code

module fork\_test;  
initial begin  
for (int j=0; j<3; j++)  
begin

fork  
automatic int k=j;  
$display(k);

join\_none

end  
end  
endmodule:fork\_test

1. Write = 50 data per 100 Clocks (alternate clock cycle), Read = 50 data per 100 Clocks (starting from 50th cycle) Find out the FIFO depth.
2. What is the output of the code

program bus\_contention;

semaphore sbus = new(1);

task block\_bus(string name, integer busval);

$display("[%0t]: %s gets key and is going to block the shared bus for 50 time units to write the value %0d", $time, name, busval);

#50;

endtask

initial

begin

fork

begin

#55;

$display("[%0t]: CPU1 requests key", $time);

sbus.get(1);

block\_bus("CPU1",1000);

$display("[%0t]: CPU1 returns key", $time);

sbus.put(1); #10 $display("[%0t]: CPU1 requests key", $time);

sbus.get(1);

block\_bus("CPU1",3000);

$display("[%0t]: CPU1 returns key", $time);

sbus.put(1);

end

begin

#25;

$display("[%0t]: CPU2 requests key", $time);

sbus.get(1);

block\_bus("CPU2",2000);

$display("[%0t]: CPU2 returns key", $time);

sbus.put(1);

#25;

$display("[%0t]: CPU2 requests key", $time);

sbus.get(1);

block\_bus("CPU2",4000);

$display("[%0t]: CPU2 returns key", $time);

sbus.put(1);

end

join

#25;

$display("[%0t]: FINISHED Simulation", $time);

end

endprogram

1. How many bins will be created and which bin will get hit in below example

Program main;

bit [0:3] y;

bit [0:2] values[$]= '{3,3,3,4,4};

covergroup cg;

cover\_point\_y : coverpoint y { bins trans\_3 = (3[\*5]); bins trans\_4 = (4[\*2]); }

endgroup

cg cg\_inst = new();

 initial foreach(values[i]) begin y = values[i];

cg\_inst.sample();

end

 endprogram

1. how many bins will be created for following covergroup

int i,j;

covergroup ct;

coverpoint i { bins i[] = { [0:1] }; }

coverpoint j { bins j[] = { [0:1] }; }

x1: cross i,j;

x2: cross i,j {

bins i\_zero = binsof(i) intersect { 0 };

}

endgroup