**Chapter -5: OOPS Classes**

Q1. Declare a class with a constructor having fields- 16-bit address 32-bit write data, 32-bit read data, 1-bit read, 1-bit write. Create an object for the class and generate 10 bits sets of values and display?

Ans. // Code your testbench here

// or browse Examples

class packet;

//class properties

bit [31:0] read\_data ;

bit [31:0] write\_data;

bit [15:0] addr;

bit write,read;

string pkt\_type;

//constructor

function new();

this.read\_data = 32'h0;

this.write\_data = 32'h0;

this.addr = 16'h0;

this.write = 0;

this.read =0;

this.pkt\_type = "PKT\_read";

endfunction

function void read1(bit [31:0] read\_data ,bit [31:0] write\_data,bit [15:0] addr,bit write,read);

this.read\_data = read\_data;

this.write\_data = write\_data;

this.addr = addr;

this. write = write;

this.read =read;

this.pkt\_type = "PKT\_write";

endfunction

function void write1(bit [31:0] read\_data ,bit [31:0] write\_data,bit [15:0] addr,bit write,read);

this.read\_data = read\_data;

this.write\_data = write\_data;

this.addr = addr;

this. write = write;

this.read =read;

this.pkt\_type = "PKT\_intialize";

endfunction

//method to display class prperties

function void display();

$display("---------------------------------------------------------");

$display("\t write\_data = %0d",write\_data);

$display("\t read\_data = %0h",read\_data);

$display("\t addr = %0h",addr);

$display("\t write = %0d",write);

$display("\t pkt\_type = %0s",pkt\_type);

$display("\t read= %0d",read);

$display("---------------------------------------------------------");

endfunction

endclass

module sv\_constructor;

packet pkt;

initial begin

repeat(5) begin

pkt = new();

pkt.display();

pkt.read1(32'habcd\_0000,32'h2345\_6789,16'h0012,0,1);

pkt.display();

pkt.write1(32'habcd\_0000,32'h2345\_6789,16'h0012,1,0);

pkt.display();

end

end

endmodule

===========================output==========================

ncsim> run

---------------------------------------------------------

write\_data = 0

read\_data = 0

addr = 0

write = 0

pkt\_type = PKT\_read

read= 0

---------------------------------------------------------

---------------------------------------------------------

write\_data = 591751049

read\_data = abcd0000

addr = 12

write = 0

pkt\_type = PKT\_write

read= 1

---------------------------------------------------------

---------------------------------------------------------

write\_data = 591751049

read\_data = abcd0000

addr = 12

write = 1

pkt\_type = PKT\_intialize

read= 0

---------------------------------------------------------

---------------------------------------------------------

write\_data = 0

read\_data = 0

addr = 0

write = 0

pkt\_type = PKT\_read

read= 0

---------------------------------------------------------

---------------------------------------------------------

write\_data = 591751049

read\_data = abcd0000

addr = 12

write = 0

pkt\_type = PKT\_write

read= 1

---------------------------------------------------------

---------------------------------------------------------

write\_data = 591751049

read\_data = abcd0000

addr = 12

write = 1

pkt\_type = PKT\_intialize

read= 0

---------------------------------------------------------

---------------------------------------------------------

write\_data = 0

read\_data = 0

addr = 0

write = 0

pkt\_type = PKT\_read

read= 0

---------------------------------------------------------

---------------------------------------------------------

write\_data = 591751049

read\_data = abcd0000

addr = 12

write = 0

pkt\_type = PKT\_write

read= 1

---------------------------------------------------------

---------------------------------------------------------

write\_data = 591751049

read\_data = abcd0000

addr = 12

write = 1

pkt\_type = PKT\_intialize

read= 0

---------------------------------------------------------

---------------------------------------------------------

write\_data = 0

read\_data = 0

addr = 0

write = 0

pkt\_type = PKT\_read

read= 0

---------------------------------------------------------

---------------------------------------------------------

write\_data = 591751049

read\_data = abcd0000

addr = 12

write = 0

pkt\_type = PKT\_write

read= 1

---------------------------------------------------------

---------------------------------------------------------

write\_data = 591751049

read\_data = abcd0000

addr = 12

write = 1

pkt\_type = PKT\_intialize

read= 0

---------------------------------------------------------

---------------------------------------------------------

write\_data = 0

read\_data = 0

addr = 0

write = 0

pkt\_type = PKT\_read

read= 0

---------------------------------------------------------

---------------------------------------------------------

write\_data = 591751049

read\_data = abcd0000

addr = 12

write = 0

pkt\_type = PKT\_write

read= 1

---------------------------------------------------------

---------------------------------------------------------

write\_data = 591751049

read\_data = abcd0000

addr = 12

write = 1

pkt\_type = PKT\_intialize

read= 0

---------------------------------------------------------

ncsim: \*W,RNQUIE: Simulation is complete.

Q2. Extended the class declared above to initialize the address and data\_in with value 0Xffff and 0x5555\_5555 respectively?

Ans. // Code your testbench here

// or browse Examples

// Code your testbench here

// or browse Examples

class packet;

//class properties

bit [31:0] read\_data ;

bit [31:0] write\_data;

bit [15:0] addr;

bit write,read;

string pkt\_type;

//constructor

function new();

this.read\_data = 32'h0;

this.write\_data = 32'h5555\_5555;

this.addr = 16'hFFFF;

this.write = 0;

this.read =0;

this.pkt\_type = "PKT\_ intialize ";

endfunction

function void read1(bit [31:0] read\_data ,bit [31:0] write\_data,bit [15:0] addr,bit write,read);

this.read\_data = read\_data;

this.write\_data = write\_data;

this.addr = addr;

this. write = write;

this.read =read;

this.pkt\_type = "PKT\_write";

endfunction

function void write1(bit [31:0] read\_data ,bit [31:0] write\_data,bit [15:0] addr,bit write,read);

this.read\_data = read\_data;

this.write\_data = write\_data;

this.addr = addr;

this. write = write;

this.read =read;

this.pkt\_type = "PKT\_read";

endfunction

//method to display class prperties

function void display();

$display("---------------------------------------------------------");

$display("\t write\_data = %0d",write\_data);

$display("\t read\_data = %0h",read\_data);

$display("\t addr = %0h",addr);

$display("\t write = %0d",write);

$display("\t pkt\_type = %0s",pkt\_type);

$display("\t read= %0d",read);

$display("---------------------------------------------------------");

endfunction

endclass

module sv\_constructor;

packet pkt;

initial begin

repeat(1) begin

pkt = new();

pkt.display();

pkt.read1(32'habcd\_0000,32'h2345\_6789,16'h0012,0,1);

pkt.display();

pkt.write1(32'habcd\_0000,32'h2345\_6789,16'h0012,1,0);

pkt.display();

end

end

endmodule

========================output==========================

xcelium> run  
---------------------------------------------------------  
 write\_data = 1431655765  
 read\_data = 0  
 addr = ffff  
 write = 0  
 pkt\_type = PKT\_ intialize   
 read= 0  
---------------------------------------------------------  
---------------------------------------------------------  
 write\_data = 591751049  
 read\_data = abcd0000  
 addr = 12  
 write = 0  
 pkt\_type = PKT\_write  
 read= 1  
---------------------------------------------------------  
---------------------------------------------------------  
 write\_data = 591751049  
 read\_data = abcd0000  
 addr = 12  
 write = 1  
 pkt\_type = PKT\_read  
 read= 0  
---------------------------------------------------------  
xmsim: \*W,RNQUIE: Simulation is complete.  
xcelium> exit

Q3. Declare a virtual packet and a virtual method to display values? Extend and add 32-bit header, 64-bit data and 16-bit CRC field?

Ans. virtual class packet;

bit[31:0] header;

bit [63:0] data;

virtual function void display();

endfunction

endclass

class extend\_pkt extends packet;

bit [15:0] crc;

function void display();

$display("crc:%d",crc);

$display("header:%d",header);

$display("data:%d",data);

endfunction

endclass

module tb();

initial begin

extend\_pkt ex\_pkt;

ex\_pkt=new();

ex\_pkt.header=32'habcd\_abcd;

ex\_pkt.data=64'habcd\_abcd\_1234\_0000;

ex\_pkt.crc=16'hab10;

ex\_pkt.display();

end

endmodule

==========================output============================xcelium> run

crc:43792

header:2882382797

data:12379739847973404672

xmsim: \*W,RNQUIE: Simulation is complete.

Q4.Give an Example for static method and static properties with a code example?

Ans. class packet;

//static property to keep track of number of pkt's created

static byte no\_of\_pkts\_created;

//constructor

function new();

//incrementing pkt count on creating an object

no\_of\_pkts\_created++;

endfunction

//Static method to display class prperties

static function void display\_packets\_created();

$display("--------------------------------------");

$display("\t %0d packets created.",no\_of\_pkts\_created);

$display("--------------------------------------");

endfunction

endclass

module static\_properties;

packet pkt[3];

initial begin

foreach(pkt[i]) begin

pkt[i] = new();

end

pkt[0].display\_packets\_created();

end

endmodule

==========================output===========================

xcelium> run

--------------------------------------

3 packets created.

--------------------------------------

xmsim: \*W,RNQUIE: Simulation is complete.

========================================================