**Chapter -6:Randomization**

Q1.Declare a class with a constructor having fields – 16 bit- address 32-bit write, 32-bit read,1 bit read,1 bit write.

1. Create an object for the class and generate 10 sets of values using randomization and display the values.

Ans. // Code your testbench here

// or browse Examples

class bus;

rand bit [15:0] addr;

rand bit[31:0] datain;

rand bit[31:0] dataout;;

rand bit read;

rand bit write;

function new();

endfunction

endclass

module ranomize();

bus b1=new;

initial begin

repeat(10) begin

if(b1.randomize()==1)

$display("addr=%h,datain=%h,dataout=%h,read=%h,write=%h",b1.addr,b1.datain,b1.dataout,b1.read,b1.write);

else

$display("randomization failed");

end

end

endmodule

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

xcelium> run  
addr=1b49,datain=3f279ac7,dataout=ecbe44ca,read=1,write=1  
addr=7cb7,datain=5eb3e7f6,dataout=30e9fa52,read=1,write=1  
addr=ec2a,datain=d3f14f9a,dataout=cadd542c,read=1,write=0  
addr=31a5,datain=f13d2769,dataout=59d3d38a,read=0,write=0  
addr=98d4,datain=4c031441,dataout=78645207,read=0,write=0  
addr=3257,datain=e9eeff67,dataout=50489e3a,read=1,write=0  
addr=d8ce,datain=31b71de7,dataout=8626e3df,read=0,write=1  
addr=95c5,datain=cb87123b,dataout=fa02d418,read=0,write=1  
addr=e405,datain=7b8d7ad8,dataout=591fe58d,read=1,write=1  
addr=8014,datain=a96674ee,dataout=1bd75a2d,read=0,write=0  
xmsim: \*W,RNQUIE: Simulation is complete.

1. Add a constructor to generate the address, which is half word aligned(16-bit)?

Ans. // Code your testbench here

// or browse Examples

class bus;

rand bit [15:0] addr;

rand bit[31:0] datain;

rand bit[31:0] dataout;;

rand bit read;

rand bit write;

constraint word\_align{addr[0]==2'b0;}

function new();

endfunction

endclass

module ranomize();

bus b1=new;

initial begin

repeat(10) begin

if(b1.randomize()==1)

$display("addr=%h,datain=%h,dataout=%h,read=%h,write=%h",b1.addr,b1.datain,b1.dataout,b1.read,b1.write);

else

$display("randomization failed");

end

end

endmodule

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

xcelium> run  
addr=4098,datain=a7b81b49,dataout=3f279ac7,read=0,write=1  
addr=2bf8,datain=45877cb7,dataout=5eb3e7f6,read=0,write=1  
addr=5b76,datain=1ae4ec2a,dataout=d3f14f9a,read=0,write=1  
addr=09f8,datain=1d3331a5,dataout=f13d2769,read=0,write=0  
addr=fcbe,datain=f7f698d4,dataout=4c031441,read=1,write=0  
addr=3a06,datain=cdd73257,dataout=e9eeff67,read=0,write=1  
addr=af40,datain=a043d8ce,dataout=31b71de7,read=1,write=0  
addr=107c,datain=13d495c5,dataout=cb87123b,read=0,write=0  
addr=cc4c,datain=d856e405,dataout=7b8d7ad8,read=1,write=1  
addr=9338,datain=a5ec8014,dataout=a96674ee,read=1,write=0  
xmsim: \*W,RNQUIE: Simulation is complete.  
xcelium> exit

1. Extend the class declared above to initialize the address and data\_in with values 0xffff and 0x55555\_5555 respectively?

Ans. // Code your testbench here

// or browse Examples

class bus;

rand bit [15:0] addr;

rand bit[31:0] datain;

rand bit[31:0] dataout;;

rand bit read;

rand bit write;

constraint word\_align{addr[1:0]==2'b00;}

function new();

this.addr=16'hffff;

this.datain=32'h5555\_5555;

this.dataout=32'h0;

this.read=0;

this.write=0;

endfunction

endclass

module ranomize();

bus b1=new;

initial begin

repeat(10) begin

if(b1.randomize()==1)

$display("addr=%h,datain=%h,dataout=%h,read=%h,write=%h",b1.addr,b1.datain,b1.dataout,b1.read,b1.write);

else

$display("randomization failed");

end

end

endmodule

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

xcelium> run  
addr=4098,datain=a7b81b49,dataout=3f279ac7,read=0,write=1  
addr=2bf8,datain=45877cb7,dataout=5eb3e7f6,read=0,write=1  
addr=5b74,datain=1ae4ec2a,dataout=d3f14f9a,read=0,write=1  
addr=09f8,datain=1d3331a5,dataout=f13d2769,read=0,write=0  
addr=fcbc,datain=f7f698d4,dataout=4c031441,read=1,write=0  
addr=3a04,datain=cdd73257,dataout=e9eeff67,read=0,write=1  
addr=af40,datain=a043d8ce,dataout=31b71de7,read=1,write=0  
addr=107c,datain=13d495c5,dataout=cb87123b,read=0,write=0  
addr=cc4c,datain=d856e405,dataout=7b8d7ad8,read=1,write=1  
addr=9338,datain=a5ec8014,dataout=a96674ee,read=1,write=0  
xmsim: \*W,RNQUIE: Simulation is complete.  
xcelium> exit

1. In the extended class, override the existing constraint to generate the address, which is word aligned (32-bit)?

Ans. // Code your testbench here

// or browse Examples

class bus;

rand bit [15:0] addr;

rand bit[31:0] datain;

rand bit[31:0] dataout;;

rand bit read;

rand bit write;

constraint word\_align{addr[1:0]==2'b0;}

function new();

this.addr=16'hffff;

this.datain=32'h5555\_5555;

this.dataout=32'h0;

this.read=0;

this.write=0;

endfunction

endclass

class ex\_bus extends bus;

constraint word\_align{addr[1:0]==2'b00;}

endclass

module ranomize();

ex\_bus b1=new;

initial begin

repeat(10) begin

if(b1.randomize()==1)

$display("addr=%h,datain=%h,dataout=%h,read=%h,write=%h",b1.addr,b1.datain,b1.dataout,b1.read,b1.write);

else

$display("randomization failed");

end

end

endmodule

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

xcelium> run  
addr=4098,datain=a7b81b49,dataout=3f279ac7,read=0,write=1  
addr=2bf8,datain=45877cb7,dataout=5eb3e7f6,read=0,write=1  
addr=5b74,datain=1ae4ec2a,dataout=d3f14f9a,read=0,write=1  
addr=09f8,datain=1d3331a5,dataout=f13d2769,read=0,write=0  
addr=fcbc,datain=f7f698d4,dataout=4c031441,read=1,write=0  
addr=3a04,datain=cdd73257,dataout=e9eeff67,read=0,write=1  
addr=af40,datain=a043d8ce,dataout=31b71de7,read=1,write=0  
addr=107c,datain=13d495c5,dataout=cb87123b,read=0,write=0  
addr=cc4c,datain=d856e405,dataout=7b8d7ad8,read=1,write=1  
addr=9338,datain=a5ec8014,dataout=a96674ee,read=1,write=0  
xmsim: \*W,RNQUIE: Simulation is complete.  
xcelium> exit

1. Generate the data\_in between the ranges 0x100 to 0xffff.

Ans. // Code your testbench here

// or browse Examples

class bus;

rand bit [15:0] addr;

rand bit[31:0] datain;

rand bit[31:0] dataout;

rand bit read;

rand bit write;

constraint word\_align{addr[1:0]==2'b0;}

function new();

this.addr=16'hffff;

this.datain=32'h5555\_5555;

this.dataout=32'h0;

this.read=0;

this.write=0;

endfunction

endclass

class ex\_bus extends bus;

constraint word\_align{addr[1:0]==2'b00;}

constraint datain\_range { datain inside { [0'h100:0'hFFFF]}; }

endclass

module ranomize();

ex\_bus b1=new;

initial begin

repeat(10) begin

if(b1.randomize()==1)

$display("addr=%h,datain=%h,dataout=%h,read=%h,write=%h",b1.addr,b1.datain,b1.dataout,b1.read,b1.write);

else

$display("randomization failed");

end

end

endmodule

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

xcelium> run  
addr=b88c,datain=00006f99,dataout=a7b81b49,read=1,write=0  
addr=6984,datain=000067f9,dataout=45877cb7,read=0,write=0  
addr=3900,datain=00007276,dataout=1ae4ec2a,read=0,write=0  
addr=990c,datain=000018f8,dataout=1d3331a5,read=1,write=0  
addr=4fb4,datain=0000c7be,dataout=f7f698d4,read=1,write=1  
addr=2790,datain=00002906,dataout=cdd73257,read=1,write=0  
addr=c774,datain=00008641,dataout=a043d8ce,read=1,write=1  
addr=fd54,datain=0000737d,dataout=13d495c5,read=1,write=0  
addr=cfbc,datain=0000d14d,dataout=d856e405,read=0,write=1  
addr=047c,datain=00005838,dataout=a5ec8014,read=0,write=1  
xmsim: \*W,RNQUIE: Simulation is complete.  
xcelium> exit

1. Split the whole address range into 3 groups, low, medium and high, assign weightage as 1, 2, 3 respectively to randomize values.

Ans.

1. Simulate the example by turning off constraint and randomization and observe the results?

Ans. // Code your testbench here

// or browse Examples

class bus;

rand bit [15:0] addr;

rand bit[31:0] datain;

rand bit[31:0] dataout;

rand bit read;

int cycles = 1;

rand bit write;

constraint word\_align{addr[1:0]==2'b00;}

function void post\_randomize();

$display("post randomize vlaue addr=%h,datain=%h,dataout=%h,read=%h,write=%h",addr,datain,dataout,read,write );

endfunction

extern function exercise;

endclass

function bus::exercise;

int res;

begin

word\_align.constraint\_mode(0);

$display("word allign address constraint disable");

repeat(cycles) begin

res=randomize();

end

$display("word allign address constraint ensable");

word\_align.constraint\_mode(1);

repeat(cycles) begin

res=randomize();

end

end

endfunction

module ranomize();

bus b1=new;

initial begin

$display("randomize mode enabled");

b1.rand\_mode(1);

b1.exercise();

$display("randomize mode disabled");

b1.rand\_mode(0);

b1.exercise();

end

endmodule

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

xcelium> run  
randomize mode enabled  
word allign address constraint disable  
post randomize vlaue addr=1b49,datain=3f279ac7,dataout=ecbe44ca,read=1,write=1  
word allign address constraint ensable  
post randomize vlaue addr=2bf8,datain=45877cb7,dataout=5eb3e7f6,read=0,write=1  
randomize mode disabled  
word allign address constraint disable  
post randomize vlaue addr=2bf8,datain=45877cb7,dataout=5eb3e7f6,read=0,write=1  
word allign address constraint ensable  
post randomize vlaue addr=2bf8,datain=45877cb7,dataout=5eb3e7f6,read=0,write=1  
xmsim: \*W,RNQUIE: Simulation is complete.  
xcelium> exit