**Chapter -2: System Verilog Data Types**

Q1.Write a system Verilog program to define an event and detect an event in other process?

Ans.

module tb();

event a;

initial begin

#10->a;

$display("thread 1 event is trrigerwed ",$time);

end

initial begin

$display("thread 2 event is trrigerwed waited start",$time);

#5 @(a);

$display("thread 2 event is trrigerwed waited ",$time);

end

initial begin

$display("thread 3 event is trrigerwed waited start",$time);

wait(a.triggered);

$display("thread 3 event is trrigerwed waited ",$time);

end

endmodule

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

sim> run

thread 2 event is trrigerwed waited start 0

thread 3 event is trrigerwed waited start 0

thread 1 event is trrigerwed 10

thread 3 event is trrigerwed waited 10

thread 2 event is trrigerwed waited 10

Q2. Write a program to store your full name in a string and find the length of your name?

Ans. module tb();

string name = "teekam\_chand\_khandelwal";

initial

$display("length of the name sting :",name.len());

endmodule

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

sim> run

length of the name sting : 23

Q3.Declare enumeration type for storing states, IDLE, SETUP, ACCESS? Assign and print the values?

Ans.

module tb();

enum {idle,access,Setup} state;

initial begin

$display("state name :%s no: %d",state.name,state);

state = state.next();

$display("state name :%s no: %d",state.name,state);

state = state.last();

$display("state name :%s no: %d",state.name,state);

end

endmodule

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

sim> run

state name :idle no: 0

state name :access no: 1

state name :Setup no: 2

Q4.Using struct, write a program to store 16-bit addr, 32-bit data\_in, 32-bit data\_out, read\_write? Assign respective values to the struct variable and print?

Ans. typedef enum logic {read,write} status;

typedef struct {

bit [15:0] addr;

bit [31:0] data\_in, data\_out;

status stts;

} mem\_data;

module tb();

mem\_data pkt;

initial begin

pkt='{16'h234a,32'haacc\_fdce,32'hx,read};

$display("addr=%d data\_in=%d dataout=%d status = %s",pkt.addr,pkt.data\_in,pkt.data\_out,pkt.stts.name);

#1 $finish;

end

endmodule

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

xcelium> run  
addr= 9034 data\_in=2865561038 dataout= 0 status = read  
Simulation complete via $finish(1) at time 1 NS + 0