Arti_tyagi_system_verilog_2023

System Verilog DataTyape

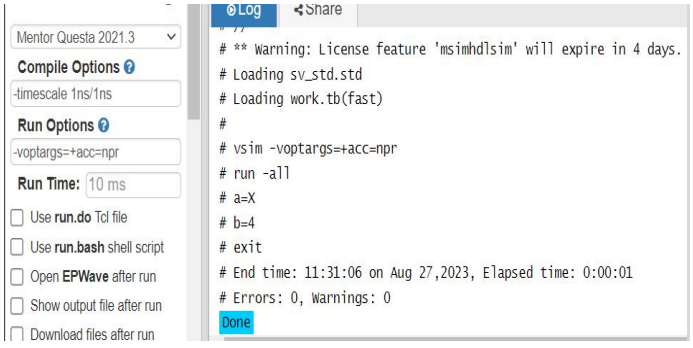
Code 1 :

Understand the difference between integer and int data type

Here integer is 4 state variables can store (0,1,x,z) and
Int is new 2 state variable can store only 0 and 1.

```
module tb;
integer [3:0] a;
int [3:0] b;

initial begin
a = 4'b01xz;
b = 4'b01xz;
$display("a=%b,b=%b",a,b);
end
endmodule
```



Code -2

Understand the difference between byte ,bit and logic data type

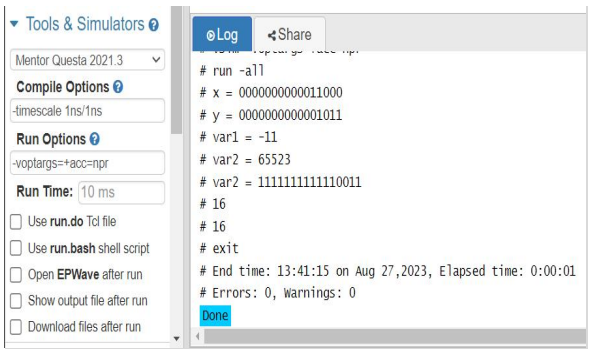
Here logic is 4 state variables can store (0,1,x,z) , bit and byte is 2 state data type which is used most often in testbench.
bit and byte can either 0 or 1 which represent single bit.

```
module tb;
  logic [4:0] a;  //declare 5 bit logic type variable
  bit b;        //1 bit variable of bit
  bit[3:0]c;
  byte d;

initial begin
            a = 4'b01xz;
   b=1;
            c = 4'b01xz;
   d = 4'b01xz;

  $display("\nlogic data types a=%b,\nbit data types b=%b,\nbit data types c=%b,\nByte data types d",a,b,c,d);

end
endmodule
```



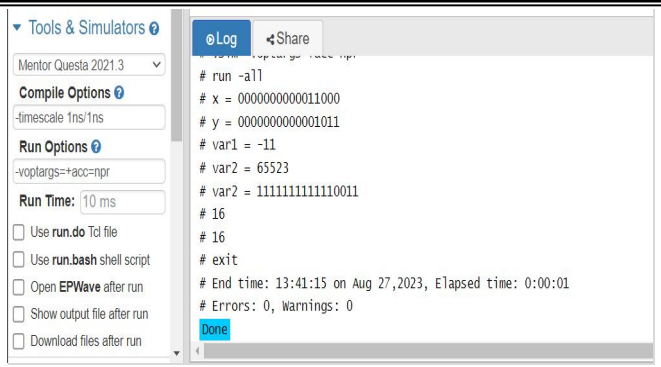
Code -3

Understand the difference between shortint ,bit and logicint data type

Shortint :- 2 state data type, 16-bit signed integer
Longint :- 2 state data type, 64-bit signed integer

```
module tb;
shortint x;//variable declaration
shortint y = 5'b01011;
shortint var1;//signed
bit [15:0] var2;//unsigned
longint var3;
int var4;

initial begin
  x = 16'b110zx;
  $display("x = %b",x);
  $display("y = %b",y);
            var1 = -11;
            var2 = -13;
            $display("var1 = %0d",var1);
            $display("var2 = %0d",var2);
            $display("var2 = %b",var2);
            $display("%0d",$size(var1));
            $display("%0d",$size(var2));
end
endmodule
```



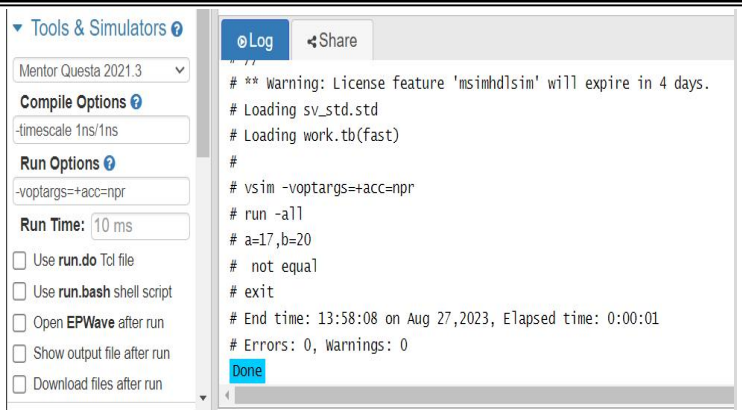
Code -4

Declare integer a,b

a. Assign a and b with random values between 10,20. display both the a,b.

```
module tb;
integer a,b;

initial begin
        a = $urandom_range(10,20);
        b = $urandom_range(10,20);
       $display("a=%0d,b=%0d",a,b);
        if(a == b)begin
                $display("equal");
                end
        else begin
                $display(" not equal");
        end
end
endmodule
```


Tools & Simulators
Mentor Questa 2021.3
Compile Options
-timescale 1ns/1ns
Run Options
-voptargs=+acc=npr
Run Time:
10 ms
Use run.do Tcl file
Use run.bash shell script
Open EPWave after run
Show output file after run
Download files after run
Log
Share
** Warning: License feature 'msimhdlsim' will expire in 4 days.
Loading sv_std.std
Loading work.tb(fast)
#
vsim -voptargs=+acc=npr
run -all
a=17,b=20
not equal
exit
End time: 13:58:08 on Aug 27,2023, Elapsed time: 0:00:01
Errors: 0, Warnings: 0
Done