// A test bench for the scrambled eggs assignment

// This test bench developed for group 14

//

`timescale 1ns/10ps

module top();

reg clk,rst;

integer cfile;

reg [31:0] case\_addr,case\_data,case\_scramble,case\_expected,case\_entrophy;

string sin;

reg [11:0] addr;

reg [31:0] wdata;

reg write;

reg res;

reg pushin;

reg [7:0] datain;

reg [31:0] entrophy;

reg [31:0] q[$];

wire pushout;

wire [31:0] dataout;

reg zero\_pushout;

reg [31:0] zero\_dataout,exp;

string dstr;

int timer;

initial begin

cfile = $fopen("cases14.txt","r");

if(cfile == 0) begin

$display("Could not open cases14.txt Simulation failed to start");

$finish;

end

end

initial begin

clk=0;

forever #5 clk=~clk;

end

default clocking cb @(posedge(clk));

endclocking

initial begin

$dumpfile("se14.vcd");

$dumpvars(9,top);

end

initial begin

rst=0;

write=0;

wdata=0;

addr=0;

pushin=0;

datain=0;

entrophy=0;

timer=100;

rst=1;

##3 #3 rst=0;

end

task failit(input string str);

begin

$display("\n\n\n\n\n");

$display("=============================================================");

$display(str);

$display("=============================================================");

$display("\n\nSimulation failed\n\n");

#10;

$finish;

end

endtask

always @(posedge(clk)) begin

if(!rst) begin

if(q.size() > 0) begin

zero\_pushout = pushout;

zero\_dataout = dataout;

#0.1;

if(zero\_pushout === 1'bx) failit("pushout is X");

if(zero\_pushout !== pushout) failit("No hold time on pushout");

if(zero\_dataout !== dataout) failit("No hold time on dataout");

if(pushout) begin

exp = q.pop\_front();

if(exp !== dataout) failit($sformatf("Received data error --- expected %h received %h",exp,dataout));

timer=100;

end else begin

timer = timer -1;

if(timer <= 0) begin

failit("I waited 100 clocks and you didn't push any result");

end

end

end else begin

timer=100;

if(pushout === 1'bx) failit("Pushout is an X");

if(pushout !== 1'b0) failit("You pushed and I don't expect data");

end

end

end

initial begin

##10 #1;

while( $fgets(sin,cfile)) begin

if(sin[0] == "#") continue;

if(sin[0] == "w") begin

res=$sscanf(sin,"%s %h %h",dstr,case\_addr,case\_data);

addr = case\_addr;

wdata = case\_data;

write = 1;

##1 #1;

write = 0;

if( ($random &32'hff) > 128) begin

wdata = 32'h12345678;

repeat ($random&15) ##1 #1;

end

end else if(sin[0]=="p") begin

##100 #1;

end else if(sin[0]=="s") begin

res=$sscanf(sin,"%s %h %h %h",dstr,case\_data,case\_entrophy,case\_expected);

pushin=1;

datain=case\_data;

entrophy = case\_entrophy;

q.push\_back(case\_expected);

##1 #1;

pushin=0;

datain=8'h5a;

entrophy=32'h87654321;

if( ($random&32'hff)> 200) begin

repeat( $random&32'h3) ##1 #1;

end

end else begin

$display("Corrupted case file");

$finish;

end

end

##100;

$display("End of the run");

$finish;

end

se14 se(clk,rst,write,addr,wdata,pushin,datain,entrophy,pushout,dataout);

endmodule