// Testbench UART

module test;

reg rxclk;

reg reset;

reg txclk;

reg load\_data\_tx;

reg [7:0] tx\_data;

reg uart\_enable;

reg tx\_out;

reg tx\_empty;

reg tx\_over\_run;

reg parity\_even\_odd;

reg read\_rx\_data;

reg [7:0] rx\_data;

reg rx\_empty;

reg receive\_frame\_error;

reg receive\_parity\_error;

reg rx\_over\_run;

reg rx\_in ;

reg [5:0] temp;

reg [7:0] rx\_reg;

reg [3:0] rx\_sample\_cnt;

reg rx\_busy;

integer i;

// Instantiate design under test

uart\_rx\_tx UART (

.reset (reset),

.load\_data\_tx (load\_data\_tx),

.tx\_data (tx\_data),

.uart\_enable (uart\_enable),

.tx\_out (tx\_out),

.tx\_empty (tx\_empty),

.tx\_over\_run (tx\_over\_run),

.parity\_even\_odd (parity\_even\_odd),

.rxclk (rxclk),

.read\_rx\_data (read\_rx\_data),

.rx\_data (rx\_data),

.rx\_in (tx\_out),//(rx\_in),

.rx\_empty (rx\_empty),

.receive\_frame\_error\_p (receive\_frame\_error),

.receive\_parity\_error\_p(receive\_parity\_error),

.rx\_over\_run\_p (rx\_over\_run),

.txclk\_chk (txclk),

.rx\_reg\_P (rx\_reg),

.rx\_sample\_cnt\_P (rx\_sample\_cnt) ,

.rx\_busy\_P (rx\_busy)

);

initial begin

// Dump waves

$dumpfile("dump.vcd");

$dumpvars(1);

$display("Reset flop.");

reset = 1;

load\_data\_tx = 0;

parity\_even\_odd = 1; // even parity

tx\_data = 8'h00;

uart\_enable = 1;

read\_rx\_data = 0;

#100;

$display("%t Release reset.", $time);

reset = 0;

#100

//first tx pkt

$display("%t first packet started", $time);

tx\_data = 8'hAB;

load\_data\_tx = 1'b1;

#3250;

load\_data\_tx = 1'b0;

#50000;

$display("%t first packet fully transmitted", $time);

$display("%t Reading first packet",$time);

read\_rx\_data = 1;

#210

read\_rx\_data = 0;

$display("%t first transmitted packet was %0h", $time, tx\_data);

$display("%t first recevied packet %0h", $time, rx\_data);

if (rx\_data == tx\_data)

$display("PASS: Transmitted and received packets are same");

else

$display("FAIL: Transmitted and received packets are not same");

//second tx pkt

$display("%t second packet started", $time);

tx\_data = 8'h65;

load\_data\_tx = 1'b1;

#3250;

load\_data\_tx = 1'b0;

#20000;

$display("%t sending thirdt pkt to be ignored.", $time);

tx\_data = 8'h66;

load\_data\_tx = 1'b1;

#3250;

load\_data\_tx = 1'b0;

#70000;

$display("%t Reading second packet",$time);

read\_rx\_data = 1;

#210

read\_rx\_data = 0;

#200;

$display("%t second transmitted packet was h65", $time);

$display("%t second recevied packet %0h", $time, rx\_data);

if (rx\_data == 8'h65)

$display("PASS: Transmitted and received packets are same");

else

$display("FAIL: Transmitted and received packets are not same");

$display("%t fourth packet started", $time);

tx\_data = 8'h34;

load\_data\_tx = 1'b1;

#3250;

load\_data\_tx = 1'b0;

#50000;

$display("%t fourth packet fully transmitted", $time);

$display("%t Reading third received packet",$time);

read\_rx\_data = 1;

#210

read\_rx\_data = 0;

#200;

$display("%t fourth transmitted packet was %0h", $time, tx\_data);

$display("%t third recevied packet %0h", $time, rx\_data);

if (rx\_data == tx\_data)

$display("PASS: Transmitted and received packets are same");

else

$display("FAIL: Transmitted and received packets are not same");

end

initial begin

$display("Toggle clk started.");

for (i = 0; i < 2000;i = i+1) begin

rxclk = 1;

#100;

rxclk = 0;

#100;

end

end

// task display;

// #1 $display("d:%0h, q:%0h, qb:%0h",

// d, q, qb);

// endtask

endmodule