`timescale 1ns/1ns

module tb\_fifo\_sync();

parameter FIFO\_DEPTH = 8;

parameter DATA\_WIDTH = 32;

reg clk = 0;

reg rst\_n;

reg cs;

reg wr\_en;

reg rd\_en;

reg [DATA\_WIDTH-1:0] data\_in;

wire [DATA\_WIDTH-1:0] data\_out;

wire empty;

wire full;

integer i;

fifo\_sync

#(.FIFO\_DEPTH(FIFO\_DEPTH),

.DATA\_WIDTH(DATA\_WIDTH))

dut

(.clk (clk ),

.rst\_n (rst\_n ),

.cs (cs ),

.wr\_en (wr\_en ),

.rd\_en (rd\_en ),

.data\_in (data\_in ),

.data\_out(data\_out),

.empty (empty ),

.full (full ));

always begin #5 clk = ~clk; end

task write\_data(input [DATA\_WIDTH-1:0] d\_in);

begin

@(posedge clk);

cs = 1; wr\_en = 1;

data\_in = d\_in;

$display($time, " write\_data data\_in = %0d", data\_in);

@(posedge clk);

cs = 1; wr\_en = 0;

end

endtask

task read\_data();

begin

@(posedge clk);

cs = 1; rd\_en = 1;

@(posedge clk);

$display($time, " read\_data data\_out = %0d", data\_out);

cs = 1; rd\_en = 0;

end

endtask

initial begin

#1;

rst\_n = 0; rd\_en = 0; wr\_en = 0;

@(posedge clk)

rst\_n = 1;

$display($time, "\n SCENARIO 1");

write\_data(1);

write\_data(10);

write\_data(100);

read\_data();

read\_data();

read\_data();

$display($time, "\n SCENARIO 2");

for (i=0; i<FIFO\_DEPTH; i=i+1) begin

write\_data(2\*\*i);

read\_data();

end

$display($time, "\n SCENARIO 3");

for (i=0; i<=FIFO\_DEPTH; i=i+1) begin

write\_data(2\*\*i);

end

for (i=0; i<FIFO\_DEPTH; i=i+1) begin

read\_data();

end

#40 $finish;

end

initial begin

$dumpfile("dump.vcd"); $dumpvars;

end

endmodule