**FIFO**

module para\_fifo #(

parameter DEPTH = 32,

parameter DATA\_W = 8)

(

input wire clk,

input wire reset,

input wire w\_en,

input wire r\_en,

input wire [DATA\_W-1:0]data\_in,

output wire [DATA\_W-1:0]data\_out,

output wire empty,

output wire full

);

logic [DEPTH-1:0][DATA\_W-1:0]fifo\_mem;

parameter PTR\_W = $clog2(DEPTH);

logic [PTR\_W-1:0]wr\_ptr;

logic [PTR\_W-1:0]rd\_ptr;

logic [DATA\_W-1:0] fifo\_pop\_data;

always\_ff @(posedge clk or posedge reset)

begin

if(reset)begin

rd\_ptr <= {PTR\_W+1{1'b0}};

wr\_ptr <= {PTR\_W+1{1'b0}};

fifo\_pop\_data <={DATA\_W+1{1'b0}};

for(int i=0; i<32; i++) begin

fifo\_mem[i] <= {DATA\_W+1{1'b0}};

end

end

else

begin

if((w\_en == 1'b1) && (full == 1'b0))

begin

fifo\_mem[wr\_ptr]<=data\_in;

wr\_ptr <= wr\_ptr + 1'b1;

end

if((r\_en == 1'b1) && (empty == 1'b0))

begin

fifo\_pop\_data <= fifo\_mem[rd\_ptr];

rd\_ptr <= rd\_ptr + 1'b1;

end

end

end

assign data\_out = fifo\_pop\_data;

assign empty = ((wr\_ptr - rd\_ptr) == 0)? 1'b1:1'b0;

assign full = ((wr\_ptr - rd\_ptr) == 31)? 1'b1:1'b0;

endmodule

**Test Bench Code**

module tb\_para\_fifo();

parameter DATA\_W = 8;

parameter DEPTH = 32;

logic clk;

logic reset;

logic w\_en;

logic[DATA\_W-1:0] data\_in;

logic r\_en;

logic[DATA\_W-1:0] data\_out;

logic full;

logic empty;

// Instantiate the RTL

para\_fifo #(.DEPTH(DEPTH), .DATA\_W(DATA\_W)) Day19(.\*);

always begin

clk = 1'b0;

#5;

clk = 1'b1;

#5;

end

initial begin

reset = 1'b1;

w\_en = 1'b0;

r\_en = 1'b0;

@(posedge clk);

reset = 1'b0;

@(posedge clk);

@(posedge clk);

for (int i =0; i<DEPTH; i++)begin

w\_en = 1'b1;

data\_in = $urandom\_range(0,2\*\*DATA\_W-1);

@(posedge clk);

end

w\_en = 1'b0;

@(posedge clk);

@(posedge clk);

for (int i =0; i<DEPTH; i++)begin

r\_en = 1'b1;

@(posedge clk);

end

r\_en <= 1'b0;

@(posedge clk);

@(posedge clk);

w\_en <= 1'b1;

data\_in <= $urandom\_range(0, 2\*\*DATA\_W-1);

@(posedge clk);

w\_en <= 1'b0;

// Push and pop both

for (int i=0; i<DEPTH; i++) begin

w\_en <= 1'b1;

r\_en <= 1'b1;

data\_in <= $urandom\_range(0, 2\*\*DATA\_W-1);

@(posedge clk);

end

w\_en <= 1'b0;

r\_en <= 1'b0;

@(posedge clk);

@(posedge clk);

$finish();

end

// Dump vcd

initial begin

$dumpfile("day19.vcd");

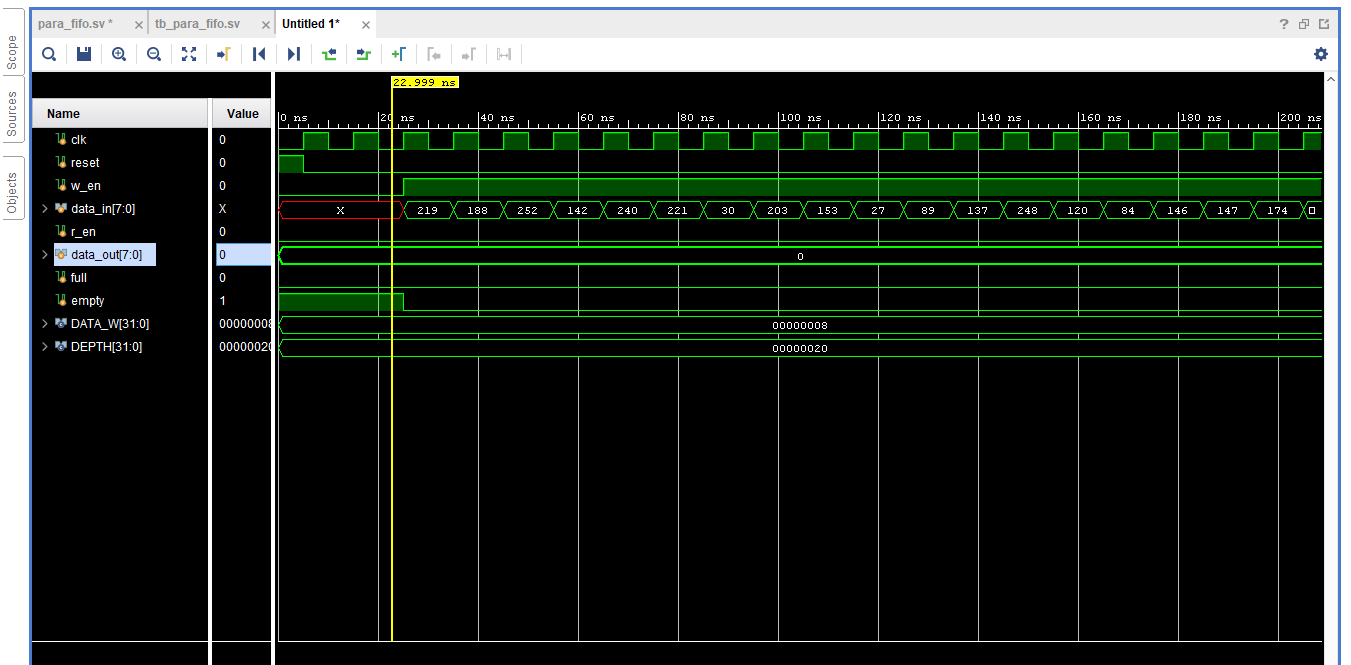
$dumpvars(0, tb\_para\_fifo);

end

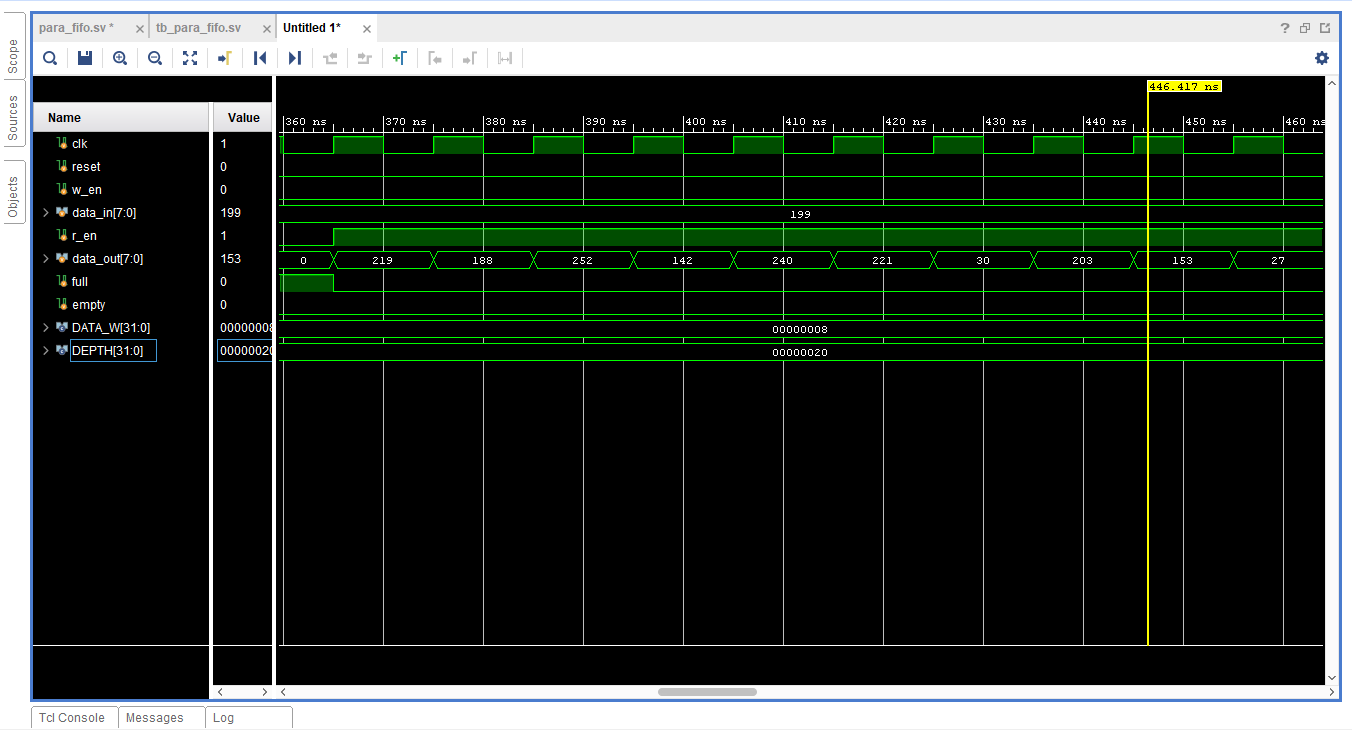
endmodule

**Simulation**

**W\_en**



r\_en



Both w\_en and r\_en