`timescale 1ns / 1ps

/\*\*

\* counter: a generic clearable up-counter

\*/

module counter

#(parameter WIDTH=64, NAME="world")

(

input clk,

input ce,

input arst\_n,

output reg [WIDTH-1:0] q

);

string name = "counter";

localparam val0 = 12'ha1f;

localparam val1 = 12'h1fa;

localparam val2 = 12'hfa1;

// some child

clock\_buffer #(WIDTH) buffer\_inst (

.clk(clk),

.ce(ce),

.reset(arst\_n)

);

// Simple gated up-counter with async clear

always @(posedge clk or negedge arst\_n) begin

if (arst\_n == 1'b0) begin

q <= {WIDTH {1'b0}};

end

else begin

q <= q;

if (ce == 1'b1) begin

q <= q + 1;

end

end

end

function int add\_one(int x);

return x + 1;

endfunction : add\_one

`ifdef SIMULATION

initial $display("Hello %s", NAME);

`endif

endmodule : counter

class my\_data extends uvm\_data;

int x, y;

function add\_one();

x++;

y++;

endfunction : add\_one

endclass : my\_data