

```

`define MSBI 13
module simulacion_test_busqueda #(parameter NUMBER_OF_SEGMENTS=4) (
    input clk,
    input start,
    output [NUMBER_OF_SEGMENTS-1:0]finish,
    output [NUMBER_OF_SEGMENTS-1:0]idle,
    input [NUMBER_OF_SEGMENTS-1:0]vector_wait_fifo_sig,
    input [NUMBER_OF_SEGMENTS-1:0]img_wait_fifo_sig,
    output [((`MSBI+2+`MSBI)+1):0]vector_me_sig1,
    output [((`MSBI+2+`MSBI)+1):0]vector_me_sig2,
    output [((`MSBI+2+`MSBI)+1):0]vector_me_sig3,
    output [((`MSBI+2+`MSBI)+1):0]vector_me_sig4,

    output [25*NUMBER_OF_SEGMENTS:0]img_mb_sig,
    output [NUMBER_OF_SEGMENTS-1:0]img_wr_req_sig,
    output [NUMBER_OF_SEGMENTS-1:0]vector_wr_req_sig,
    output [4*NUMBER_OF_SEGMENTS:0]real_state,
    output [`MSBI*NUMBER_OF_SEGMENTS:0]_realref,
    output [`MSBI*NUMBER_OF_SEGMENTS:0]_realact,
    input [`MSBI:0]window_limit,
    input [`MSBI:0]init_ref1,
    input [`MSBI:0]limit_ref1,
    input [`MSBI:0]init_ref2,
    input [`MSBI:0]limit_ref2,
    input [`MSBI:0]init_ref3,
    input [`MSBI:0]limit_ref3,
    input [`MSBI:0]init_ref4,
    input [`MSBI:0]limit_ref4,

    output [((`MSBI+2+`MSBI)+1):0]vector_me_sig_reg1,
    output [((`MSBI+2+`MSBI)+1):0]vector_me_sig_reg2,
    output [((`MSBI+2+`MSBI)+1):0]vector_me_sig_reg3,
    output [((`MSBI+2+`MSBI)+1):0]vector_me_sig_reg4
);

reg [((`MSBI+2+`MSBI)+1):0]vector_me_sig_reg_reg[NUMBER_OF_SEGMENTS-1:0];

initial
begin

vector_me_sig_reg_reg[0][((`MSBI+2+`MSBI)+1):0] = 0;
vector_me_sig_reg_reg[1][((`MSBI+2+`MSBI)+1):0] = 0;
vector_me_sig_reg_reg[2][((`MSBI+2+`MSBI)+1):0] = 0;
vector_me_sig_reg_reg[3][((`MSBI+2+`MSBI)+1):0] = 0;

end

wire [((`MSBI+2+`MSBI)+1):0]vector_me_sig_wire[NUMBER_OF_SEGMENTS-1:0];
wire [`MSBI:0]init_ref[NUMBER_OF_SEGMENTS-1:0];
wire [`MSBI:0]limit_ref[NUMBER_OF_SEGMENTS-1:0];
wire [24:0]data_rd_img_ref_sig[NUMBER_OF_SEGMENTS-1:0];

```

```

wire [24:0]data_rd_img_Act_sig[NUMBER_OF_SEGMENTS-1:0];
wire [`MSBI:0]add_write_img_ref_sig[NUMBER_OF_SEGMENTS-1:0];
wire [`MSBI:0]add_read_img_ref_sig[NUMBER_OF_SEGMENTS-1:0];
wire [NUMBER_OF_SEGMENTS-1:0]wr_enable_ref_sig;
wire [NUMBER_OF_SEGMENTS-1:0]wr_enable_act_sig;
wire [`MSBI:0]add_read_img_act_sig[NUMBER_OF_SEGMENTS-1:0];
wire [`MSBI:0]add_write_img_act_sig[NUMBER_OF_SEGMENTS-1:0];
wire [24:0]data_wr_img_ref_sig[NUMBER_OF_SEGMENTS-1:0];
wire [24:0]data_wr_img_Act_sig[NUMBER_OF_SEGMENTS-1:0];

```

```

assign init_ref[0][`MSBI:0]=init_ref1[`MSBI:0];
assign limit_ref[0][`MSBI:0]=limit_ref1[`MSBI:0];
assign init_ref[1][`MSBI:0]=init_ref2[`MSBI:0];
assign limit_ref[1][`MSBI:0]=limit_ref2[`MSBI:0];
assign init_ref[2][`MSBI:0]=init_ref3[`MSBI:0];
assign limit_ref[2][`MSBI:0]=limit_ref3[`MSBI:0];
assign init_ref[3][`MSBI:0]=init_ref4[`MSBI:0];
assign limit_ref[3][`MSBI:0]=limit_ref4[`MSBI:0];

```

```

assign
vector_me_sig_reg1[((`MSBI+2+`MSBI)+1):0]=vector_me_sig_reg_reg[0][((`MSBI+2+`MSBI)+1):0];

```

```

assign
vector_me_sig_reg2[((`MSBI+2+`MSBI)+1):0]=vector_me_sig_reg_reg[1][((`MSBI+2+`MSBI)+1):0];

```

```

assign
vector_me_sig_reg3[((`MSBI+2+`MSBI)+1):0]=vector_me_sig_reg_reg[2][((`MSBI+2+`MSBI)+1):0];

```

```

assign
vector_me_sig_reg4[((`MSBI+2+`MSBI)+1):0]=vector_me_sig_reg_reg[3][((`MSBI+2+`MSBI)+1):0];

```

```

assign
vector_me_sig1[((`MSBI+2+`MSBI)+1):0]=vector_me_sig_wire[0][((`MSBI+2+`MSBI)+1):0];

```

```

assign
vector_me_sig2[((`MSBI+2+`MSBI)+1):0]=vector_me_sig_wire[1][((`MSBI+2+`MSBI)+1):0];

```

```

assign
vector_me_sig3[((`MSBI+2+`MSBI)+1):0]=vector_me_sig_wire[2][((`MSBI+2+`MSBI)+1):0];

```

```

assign
vector_me_sig4[((`MSBI+2+`MSBI)+1):0]=vector_me_sig_wire[3][((`MSBI+2+`MSBI)+1):0];

```

```

genvar index;

```

```

generate

```

```

    for(index=0; index<NUMBER_OF_SEGMENTS; index=index+1)

```

```

    begin: estimadores

```

```

        busqueda busqueda_inst

```

```

        (

```

```

        .clk_fsm(clk) , // input clk_fsm_sig
        .start(start) , // input start_sig
        .finish(finish[index]) , // output finish_sig
        .idle(idle[index]) , // output idle_sig
        .cont_img(2'd1) , // input [1:0] cont_img_sig
        .vector_wait_fifo(vector_wait_fifo_sig[index]) ,
// input vector_wait_fifo_sig
        .img_wait_fifo(img_wait_fifo_sig[index]) , //
input img_wait_fifo_sig

        .vector_me(vector_me_sig_wire[index][(`MSBI+2+`MSBI)+1):0)) , //
        .img_mb(img_mb_sig[25*(index+1):25*index]) , //
output [25:0] img_mb_sig
        .img_wr_req(img_wr_req_sig[index]) , // output
img_wr_req_sig
        .vector_wr_req(vector_wr_req_sig[index]) , //
output vector_wr_req_sig

        .data_rd_img_ref(data_rd_img_ref_sig[index][24:0])
, // input [24:0] data_rd_img_ref_sig
        .data_rd_img_Act(data_rd_img_Act_sig[index][24:0])
, // input [24:0] data_rd_img_Act_sig

        .add_read_img_ref(add_read_img_ref_sig[index][`MSBI:0]) , // output
[10:0] add_read_img_ref_sig

        .add_write_img_ref(add_write_img_ref_sig[index][`MSBI:0]) , //
output [10:0] add_write_img_ref_sig
        .wr_enable_ref(wr_enable_ref_sig[index]) , //
output wr_enable_ref_sig

        .add_read_img_act(add_read_img_act_sig[index][`MSBI:0]) , // output
[10:0] add_read_img_act_sig

        .add_write_img_act(add_write_img_act_sig[index][`MSBI:0]) , //
output [10:0] add_write_img_act_sig
        .wr_enable_act(wr_enable_act_sig[index]) , //
output wr_enable_act_sig
        .data_wr_img_ref(data_wr_img_ref_sig[index][24:0])
, // output [24:0] data_wr_img_ref_sig
        .data_wr_img_Act(data_wr_img_Act_sig[index][24:0])
, // output [24:0] data_wr_img_Act_sig
        .window_limit(window_limit[`MSBI:0]), // input
[10:0] window_limit_sig
        .real_state(real_state[4*(index+1):4*(index)]),

        ._realact(_realact[`MSBI*(index+1):`MSBI*(index)]),

        ._realref(_realref[`MSBI*(index+1):`MSBI*(index)]),
        .init_ref(init_ref[index][`MSBI:0]), // puntero
inicial de memoria para la imagen de referencia o la imagen mas lenta
        .limit_ref(limit_ref[index][`MSBI:0])
);

```

```

        ram_simulacion_ref ram_simulacion_act_inst
        (
            .clk(clk) , // input  clk_sig
            .addr_rd(add_read_img_ref_sig[index][`MSBI:0]) ,
            // input [10:0] addr_rd_sig
            .addr_wr(add_write_img_ref_sig[index][`MSBI:0]) ,
            // input [10:0] addr_wr_sig
            .in(data_wr_img_ref_sig[index][24:0]) , // input
[24:0] in_sig
            .wr_enm(wr_enable_ref_sig[index]) ,      // input
wr_enm_sig
            .out(data_rd_img_ref_sig[index][24:0]) // output
[24:0] out_sig
        );

        ram_simulacion_act ram_simulacion_ref_inst
        (
            .clk(clk) , // input  clk_sig
            .addr_rd(add_read_img_act_sig[index][`MSBI:0]) ,
            // input [10:0] addr_rd_sig
            .addr_wr(add_write_img_act_sig[index][`MSBI:0]) ,
            // input [10:0] addr_wr_sig
            .in(data_wr_img_Act_sig[index][24:0]) , // input
[24:0] in_sig
            .wr_enm(wr_enable_act_sig[index]) ,      // input
wr_enm_sig
            .out(data_rd_img_Act_sig[index][24:0]) // output
[24:0] out_sig
        );

        always@(posedge vector_wr_req_sig[index])
        begin

            vector_me_sig_reg_reg[index][((`MSBI+2+`MSBI)+1):0]<=vector_me_sig_wire[i
ndex][((`MSBI+2+`MSBI)+1):0];
            end

        end

    endgenerate

endmodule

```