```
module router_fifo(clk,rstn,soft_rst,we,re,din,lfd_state,data_out,empty,full);
parameter depth=16,width=9,addr_bus_width=5;
input clk,rstn,soft_rst,we,re,lfd_state;
input[width-2:0]din;
output reg[width-2:0]data_out;
output full,empty;
reg[addr_bus_width-1:0]wr_pt,rd_pt;
reg fifo_counter;
reg lfd_state_s;
reg[width-1:0]mem[depth-1:0];
integer i;
always@(posedge clk)
begin
if(~rstn)
lfd_state_s<=0;
else
lfd_state_s<=lfd_state;
end
always@(posedge clk)
begin
        if(!rstn)
        begin
                for(i=0;i<16;i=i+1) begin
                        mem[i]<=0;
                end
        end
        else if(soft_rst)
        begin
                for(i=0;i<16;i=i+1) begin
                        mem[i]<=0;
                end
        end
        else
        begin
                if(we && !full)
                {mem[wr_pt[3:0]]}<={lfd_state_s,din};
end
```

```
	end
	always@(posedge clk)
	begin
		if(!rstn)
		begin
			fifo_counter<=0;
		end
		else if(soft_rst)
		begin
			fifo_counter<=0;
		end
		else if(re & ~empty)
		begin
			if(mem[rd_pt[3:0]][8]==1'b1)
				fifo_counter<=mem[rd_pt[3:0]][7:2]+1'b1;
			else if(fifo_counter!=0)
				fifo_counter<=fifo_counter-1'b1;
		end
	end
	always@(posedge clk)
	begin
		if(!rstn)
			data_out<=8'b00000000;
		else if(soft_rst)
			data_out<=8'bZZZZZZZZ;
		/*else
		begin
			if(fifo_counter==0 && data_out!=0)
				data_out<=8'dZ;*/
			else if(re && !empty)
				data_out<=mem[rd_pt[3:0]];
	//	end
	end
	always@(posedge clk)
	begin
		if(!rstn)
		begin
```

```
			rd_pt<=5'b00000;
			wr_pt<=5'b00000;
		end
		else if(soft_rst)
		begin
			rd_pt<=5'b00000;
			wr_pt<=5'b00000;
		end
		else
		begin
			if(!full && we)
				wr_pt<=wr_pt+1;
			else
				wr_pt<=wr_pt;
			if(!empty && re)
				rd_pt<=rd_pt+1;
			else
				rd_pt<=rd_pt;
		end
	end
	assign full=(wr_pt=={~rd_pt[4],rd_pt[3:0]})?1'b1:1'b0;
	assign empty=(wr_pt==rd_pt)?1'b1:1'b0;
	endmodule
```

```
module
router_synchronizer(detect_addr,write_enb_reg,clk,rstn,din,re_0,re_1,re_2,empty_0,empty_1,empty_2,full_0
,full_1,full_2,fifo_full,we,soft_rst_0,soft_rst_1,soft_rst_2,valid_out_0,valid_out_1,valid_out_2);

input detect_addr,write_enb_reg,clk,rstn;

input[1:0]din;

input re_0,re_1,re_2;

input empty_0,empty_1,empty_2;

input full_0,full_1,full_2;

output reg fifo_full;

output reg[2:0]we;

output reg soft_rst_0,soft_rst_1,soft_rst_2;

output valid_out_0,valid_out_1,valid_out_2;

reg timer_0,timer_1,timer_2;

reg int_addr_reg;

always@(posedge clk)

begin

        if(~(rstn))

begin

                timer_0<=0;

        soft_rst_0<=0;

end

else if(valid_out_0)

begin

        if(!re_0)

        begin

                if(timer_0==5'd29)

                begin

                        soft_rst_0<=1;

                        timer_0<=0;

                end

                else

                begin

                        soft_rst_0<=0;

                        timer_0<=timer_0+1'b1;

                end

        end

end

end
```

```
always@(posedge clk)
begin
        if(~(rstn))
begin
                timer_1<=0;
        soft_rst_1<=0;
end
else if(valid_out_1)
begin
        if(!re_1)
        begin
                if(timer_1==5'd29)
                begin
                        soft_rst_1<=1;
                        timer_1<=0;
                end
                else
                begin
                        soft_rst_1<=0;
                        timer_1<=timer_1+1'b1;
                end
        end
end
end
always@(posedge clk)
begin
        if(~(rstn))
begin
                timer_2<=0;
        soft_rst_2<=0;
end
else if(valid_out_2)
begin
        if(!re_2)
        begin
                if(timer_2==5'd29)
                begin
                soft_rst_2<=1;
                timer_2<=0;
            end
            else
            begin
                soft_rst_2<=0;
                timer_2<=timer_2+1'b1;
            end
        end
    end
    end
    always@(posedge clk)
    begin
        if(~rstn)
            int_addr_reg<=0;
        else if(detect_addr)
            int_addr_reg<=din;
    end
    always@(*)
    begin
        we=3'b000;
        if(write_enb_reg)
        begin
            case(int_addr_reg)
                2'b00:we=3'b001;
                2'b01:we=3'b010;
                2'b10:we=3'b100;
                default:we=3'b000;
            endcase
        end
    end
    always@(*)
    begin
        case(int_addr_reg)
            2'b00:fifo_full=full_0;
            2'b01:fifo_full=full_1;
            2'b10:fifo_full=full_2;
```

```
          endcase
end
assign valid_out_0=~empty_0;
assign valid_out_1=~empty_1;
assign valid_out_2=~empty_2;
endmodule
```

```
module
router_controller(clk,rstn,pkt_valid,parity_done,soft_rst_0,soft_rst_1,soft_rst_2,fifo_full,low_pkt_valid,fifo_
empty_0,fifo_empty_1,fifo_empty_2,

din,busy,detect_addr,lfd_state,laf_state,full_state,write_enb_reg,rst_int_reg,ld_state);

input clk,rstn,pkt_valid,parity_done,fifo_full,low_pkt_valid;

input soft_rst_0,soft_rst_1,soft_rst_2;

input fifo_empty_0,fifo_empty_1,fifo_empty_2;

input [1:0]din;

output busy,detect_addr,lfd_state,laf_state,full_state,write_enb_reg,rst_int_reg,ld_state;

parameter
decode_address=3'b000,load_first_data=3'b001,load_data=3'b010,fifo_full_state=3'b011,load_after_full=3'b
100,load_parity=3'b101,check_parity_error=3'b110,wait_till_empty=3'b111;

reg[2:0]pre,next;

reg[2:0]addr;

always@(posedge clk)

begin

if(~rstn)

pre<=next;

else

if((soft_rst_0&&din==2'b00)||(soft_rst_0&&din==2'b01)||(soft_rst_2&&din==2'b10))

pre<=decode_address;

else

pre<=next;

end

always@(*)

begin

next=pre;

begin

case(pre)

decode_address:
if((pkt_valid&&(din==0)&&fifo_empty_0)||(pkt_valid&&(din==1)&&fifo_empty_1)||(pkt_valid&&(din==2
)&&fifo_empty_2))

                    next=load_first_data;

                    else if((pkt_valid&&(din==0)&& ~fifo_empty_0)||(pkt_valid&&(din==1)&&
~fifo_empty_1)||(pkt_valid&&(din==2)&& ~fifo_empty_2))

                    next=wait_till_empty;

                    else next=load_data;

load_first_data: next=load_data;

load_data:          if(fifo_full)

                    next=fifo_full_state;
```

```
                          else if (~fifo_full&& ~pkt_valid)
                          next=load_parity;
                          else next=load_data;
fifo_full_state: if(~fifo_full)
                          next=load_after_full;
                           else next=fifo_full_state;
load_after_full: if(~parity_done&&low_pkt_valid)
                          next=load_parity;
                           else if(~parity_done&& ~low_pkt_valid)
                          next=load_data;
                           else next=decode_address;
load_parity: next=check_parity_error;
check_parity_error: if(fifo_full)
                             next=fifo_full_state;
                               else next=decode_address;
wait_till_empty:
if((fifo_empty_0&&(addr==0))||(fifo_empty_1&&(addr==1))||(fifo_empty_2&&(addr==2)))
                                        next=load_first_data;
                            else next=wait_till_empty;
default: next=decode_address;
endcase
end
end
always@(posedge clk)
begin
if(~rstn)
addr<=0;
else
if((soft_rst_0&&din==2'b00)||(soft_rst_1&&din==2'b01)||(soft_rst_2&&din==2'b10))
addr<=0;
else if(decode_address)
addr<=din;
end
assign detect_addr=(pre==decode_address)?1'b1:1'b0;
assign lfd_state=(pre==load_first_data)?1'b1:1'b0;
assign full_state=(pre==fifo_full_state)?1'b1:1'b0;
assign ld_state=(pre==load_data)?1'b1:1'b0;
assign laf_state=(pre==load_after_full)?1'b1:1'b0;
assign rst_int_reg=(pre==check_parity_error)?1'b1:1'b0;

assign write_enb_reg=((pre==load_data||pre==load_after_full||pre==load_parity))?1'b1:1'b0;

assign
busy=((pre==load_first_data||pre==fifo_full_state||pre==load_after_full||pre==load_parity||pre==check_parit
y_error||pre==wait_till_empty))?1'b1:1'b0;

endmodule
```

```
module
router_register(clk,rstn,pkt_valid,fifo_full,rst_int_reg,detect_addr,ld_state,laf_state,full_state,lfd_state,din,pa
rity_done,low_pkt_valid,error,dout);

input clk,rstn,pkt_valid,fifo_full,rst_int_reg,detect_addr,ld_state,laf_state,full_state,lfd_state;

input[7:0]din;

output reg parity_done,low_pkt_valid,error;

output reg [7:0]dout;

reg [7:0]header_byte,fifo_full_state_byte,internal_parity,pkt_parity;

always@(posedge clk)

begin

if(~rstn)

parity_done<=0;

else

begin

if(ld_state&& ~pkt_valid&& ~fifo_full)

parity_done<=1'b1;

else if(laf_state&&low_pkt_valid&& ~parity_done)

parity_done<=1'b1;

else

begin

if(detect_addr)

parity_done<=0;

end

end

end

always@(posedge clk)

begin

if(~rstn)

dout<=0;

else if(lfd_state)

dout<=header_byte;

else if(ld_state&& ~fifo_full)

dout<=din;

else if(laf_state)

dout<=fifo_full_state_byte;

else

dout<=dout;

end
```

```
always@(posedge clk)
begin
if(~rstn)
{header_byte,fifo_full_state_byte}<=0;
else
begin
if(pkt_valid&&detect_addr)
header_byte<=din;
else if(ld_state&&fifo_full)
fifo_full_state_byte<=din;
end
end
always@(posedge clk)
begin
if(~rstn)
low_pkt_valid<=1'b0;
else if(rst_int_reg)
low_pkt_valid<=1'b0;
else
begin
if(~pkt_valid && ld_state)
low_pkt_valid<=1'b1;
end
end
always@(posedge clk)
begin
if(~rstn)
pkt_parity<=0;
else if((ld_state&& ~pkt_valid&& ~fifo_full)||(laf_state&&low_pkt_valid&& ~parity_done))
pkt_parity<=din;
else if(~pkt_valid&&rst_int_reg)
pkt_parity<=0;
else
begin
if(detect_addr)
pkt_parity<=0;
end
```

```
end
always@(posedge clk)
begin
if(~rstn)
internal_parity<=8'b0;
else if(detect_addr)
internal_parity<=8'b0;
else if (lfd_state)
internal_parity<=header_byte;
else if(ld_state&&pkt_valid&& ~full_state)
internal_parity<=internal_parity^din;
else if(~pkt_valid&&rst_int_reg)
internal_parity<=0;
end
always@(posedge clk)
begin
if(~rstn)
error<=0;
else
begin
if(parity_done==1'b1 && (internal_parity!=pkt_parity))
error<=1'b1;
else
error<=0;
end
end
endmodule
```

```
module
router_top(clk,rstn,pkt_valid,re_0,re_1,re_2,din,data_out_0,data_out_1,data_out_2,valid_out_0,valid_out_1,
valid_out_2,error,busy);

input clk,rstn,pkt_valid;

input re_0,re_1,re_2;

input[7:0]din;

output error,busy;

output[7:0]data_out_0;

output[7:0]data_out_1;

output[7:0]data_out_2;

output valid_out_0,valid_out_1,valid_out_2;

wire[2:0] re;

wire[2:0] we;

wire [2:0]soft_rst;

wire [2:0]empty;

wire [2:0]full;

wire lfd_state;

wire lfd_state_s=lfd_state;

wire [7:0]dout;

wire[7:0]data_out_temp[2:0];

genvar i;

generate

for(i=0;i<3;i=i+1)

begin: fifo

router_fifo f(.clk(clk),.rstn(rstn),.we(we[i]),.soft_rst(soft_rst[i]),.re(re[i]),.
din(dout),.lfd_state(lfd_state_s),.empty(empty[i]),.full(full[i]),.data_out(data_out_temp[i]));

end

endgenerate

router_synchronizer
s1(.clk(clk),.rstn(rstn),.din(din[1:0]),.write_enb_reg(write_enb_reg),.detect_addr(detect_addr),.valid_out_0(v
alid_out_0),.valid_out_1(valid_out_1),.valid_out_2(valid_out_2),.re_0(re[0]),.re_1(re[1]),.re_2(re[2]),.full_0
(full[0]),.full_1(full[1]),.full_2(full[2]),.soft_rst_0(soft_rst[0]),.soft_rst_1(soft_rst[1]),.soft_rst_2(soft_rst[2]),
.empty_0(empty[0]),.empty_1(empty[1]),.empty_2(empty[2]),.fifo_full(fifo_full),.we(we));

router_controller
c1(.clk(clk),.rstn(rstn),.pkt_valid(pkt_valid),.busy(busy),.parity_done(parity_done),.din(din[1:0]),.soft_rst_0(
soft_rst[0]),.soft_rst_1(soft_rst[1]),.soft_rst_2(soft_rst[2]),.fifo_full(fifo_full),.fifo_empty_0(empty[0]),.fifo_
empty_1(empty[1]),.fifo_empty_2(empty[2]),.detect_addr(detect_addr),.ld_state(ld_state),.low_pkt_valid(lo
w_pkt_valid),.laf_state(laf_state),.full_state(full_state),.write_enb_reg(write_enb_reg),.rst_int_reg(rst_int_re
g),.lfd_state(lfd_state));

router_register
r1(.clk(clk),.rstn(rstn),.pkt_valid(pkt_valid),.din(din),.fifo_full(fifo_full),.rst_int_reg(rst_int_reg),.detect_add
r(detect_addr),.ld_state(ld_state),.laf_state(laf_state),.full_state(full_state),.lfd_state(lfd_state),.parity_done(p
arity_done),.low_pkt_valid(low_pkt_valid),.error(error),.dout(dout));
```

```
assign re[0]=re_0;
assign re[1]=re_1;
assign re[2]=re_2;
assign data_out_0=data_out_temp[0];
assign data_out_1=data_out_temp[1];
assign data_out_2=data_out_temp[2];
endmodule
```

```
module router_fifo_tb();
parameter depth=16,width=9,addr_bus_width=5;
reg clk,rstn,soft_rst,we,re,lfd_state;
reg[width-2:0]din;
wire[width-2:0]data_out;
wire full,empty;
reg[width-1:0]mem[depth-1:0];
integer i=0;
integer j=1;
integer k;
parameter cycle=10;
router_fifo dut(clk,rstn,soft_rst,we,re,din,lfd_state,data_out,empty,full);
always
begin
        #(cycle/2);
        clk=1'b0;
        #(cycle/2);
        clk=~clk;
end
task soft_dut();
begin
@(negedge clk);
soft_rst=1'b1;
@(negedge clk);
soft_rst=1'b0;
end
endtask
task rst_dut();
begin
@(negedge clk);
rstn=1'b0;
@(negedge clk);
rstn=1'b1;
end
endtask
task write;
        reg[7:0]payload_data,parity,header;
        reg[5:0]payload_len;
        reg[1:0]addr;
        begin
            @(negedge clk);
            payload_len=6'd14;
            addr=2'b01;
            header={payload_len,addr};
            din=header;
            lfd_state=1'b1;
            we=1;
            for(k=0;k<payload_len;k=k+1)
        begin
            @(negedge clk);
            lfd_state=0;
            payload_data={$random}%256;
            din=payload_data;
        end
        @(negedge clk);
        parity={$random}%256;
        din=parity;
    end
    endtask
    task read(input i,input j);
    begin
    @(negedge clk)
    we=i;
    re=j;
    end
    endtask
    initial
    begin
    rst_dut;
    write;
    re=i;
    din=$random;
    #500;
    we=i;
```

```
re=j;

#200;

$monitor("clk=%b,rstn=%b,soft_rst=%b,we=%b,re=%b,din=%b,lfd_state=%b,data_out=%b,empty=%b,full
=%b",clk,rstn,soft_rst,we,re,din,lfd_state,data_out,empty,full);

#100;

$finish;

end

endmodule
```

```
module router_synchronizer_tb();

reg detect_addr,write_enb_reg,clk,rstn;

reg[1:0]din;

reg re_0,re_1,re_2;

reg empty_0,empty_1,empty_2;

reg full_0,full_1,full_2;

wire fifo_full;

wire valid_out_0,valid_out_1,valid_out_2;

wire soft_rst_0,soft_rst_1,soft_rst_2;

wire [2:0]we;

parameter cycle=10;

router_synchronizer
DUT(detect_addr,write_enb_reg,clk,rstn,din,re_0,re_1,re_2,empty_0,empty_1,empty_2,full_0,full_1,full_2,f
ifo_full,we,soft_rst_0,soft_rst_1,soft_rst_2,valid_out_0,valid_out_1,valid_out_2);

always

begin

        #(cycle/2);

        clk=1'b0;

        #(cycle/2);

        clk=1'b1;

end

task rst_dut();

        begin

                @(negedge clk);

                rstn=1'b0;

                @(negedge clk);

                rstn=1'b1;

        end

endtask

task initialize;

        begin

                detect_addr=1'b0;

                din=2'b00;

                write_enb_reg=1'b0;

                {empty_0,empty_1,empty_2}=3'b111;

                {full_0,full_1,full_2}=3'b000;

                {re_0,re_1,re_2}=3'b000;

        end
```

```
endtask
task addr(input[1:0]m);
        begin
                @(negedge clk);
                detect_addr=1'b1;
                @(negedge clk);
                detect_addr=1'b0;
        end
endtask
task write;
        begin
                @(negedge clk);
                write_enb_reg=1;
                @(negedge clk);
                write_enb_reg=0;
        end
endtask
task stimulus;
        begin
                @(negedge clk);
                {full_0,full_1,full_2}=3'b001;
                @(negedge clk);
                {re_0,re_1,re_2}=3'b001;
                @(negedge clk);
                {empty_0,empty_1,empty_2}=3'b110;
        end
endtask
initial
begin
        initialize;
        rst_dut;
        addr(2'b10);
        stimulus;
        #500;
        $finish;
end
endmodule
```

```
module router_controller_tb();

reg clk,rstn,pkt_valid,parity_done,low_pkt_valid,fifo_full;

reg soft_rst_0,soft_rst_1,soft_rst_2;

reg fifo_empty_0,fifo_empty_1,fifo_empty_2;

reg[1:0] din;

wire busy,detect_addr,ld_state,laf_state,lfd_state,full_state,write_enb_reg,rst_int_reg;

parameter cycle=10;

router_controller
dut(clk,rstn,pkt_valid,parity_done,soft_rst_0,soft_rst_1,soft_rst_2,fifo_full,low_pkt_valid,fifo_empty_0,fifo
_empty_1,fifo_empty_2,din,busy,detect_addr,lfd_state,laf_state,full_state,write_enb_reg,rst_int_reg,ld_state)
;

always@(posedge clk)

begin

#(cycle/2);

clk=1'b0;

#(cycle/2);

clk=~clk;

end

task reset;

begin

@(negedge clk)

rstn=1'b0;

@(negedge clk)

rstn=1'b1;

end

endtask

task t1();

begin

@(negedge clk)

pkt_valid=1'b1;

din=2'b01;

fifo_empty_1=1'b1;

@(negedge clk)

@(negedge clk)

fifo_full=1'b0;

pkt_valid=1'b0;

@(negedge clk)

@(negedge clk)

fifo_full=1'b0;
```

```
end
endtask
task t2();
begin
@(negedge clk)
pkt_valid=1'b1;
din=2'b01;
fifo_empty_1=1'b1;
@(negedge clk)
@(negedge clk)
fifo_full=1'b1;
@(negedge clk)
fifo_full=1'b0;
@(negedge clk)
parity_done=1'b0;
low_pkt_valid=1'b1;
@(negedge clk)
@(negedge clk)
fifo_full=1'b0;
end
endtask
task t3();
begin
@(negedge clk)
pkt_valid=1'b1;
din=2'b01;
fifo_empty_1=1'b1;
@(negedge clk)
@(negedge clk)
fifo_full=1'b1;
@(negedge clk)
fifo_full=1'b0;
@(negedge clk)
parity_done=1'b0;
low_pkt_valid=1'b0;
@(negedge clk)
fifo_full=1'b0;
```

```
pkt_valid=1'b0;
@(negedge clk)
@(negedge clk)
fifo_full=1'b0;
end
endtask
task t4();
begin
@(negedge clk)
pkt_valid=1'b1;
din=2'b01;
fifo_empty_1=1'b1;
@(negedge clk)
@(negedge clk)
fifo_full=1'b0;
pkt_valid=1'b0;
@(negedge clk)
@(negedge clk)
fifo_full=1'b1;
@(negedge clk)
fifo_full=1'b0;
@(negedge clk)
parity_done=1'b1;
end
endtask
initial
begin
reset;
#20;
t1();
#40;
t2();
#40;
t3();
#40;
t4();
#40;
```

```
reset;
#100;
$finish;
end
endmodule
```

```
module router_register_tb();

reg clk,rstn,pkt_valid,fifo_full,rst_int_reg,detect_addr,ld_state,laf_state,full_state,lfd_state;

reg [7:0]din;

wire parity_done,low_pkt_valid,error;

wire [7:0]dout;

router_register
dut(clk,rstn,pkt_valid,fifo_full,rst_int_reg,detect_addr,ld_state,laf_state,full_state,lfd_state,din,parity_done,l
ow_pkt_valid,error,dout);

parameter cycle=10;

integer i;

always@(posedge clk)

begin

#(cycle/2);

clk=1'b0;

#(cycle/2);

clk=~clk;

end

task reset;

begin

@(negedge clk)

rstn=1'b0;

@(negedge clk)

rstn=1'b1;

end

endtask

task pkt_gen();

reg[7:0]payload_data,parity,header;

reg[5:0]payload_len;

reg[1:0]addr;

begin

@(negedge clk)

payload_len=6'd6;

addr=2'b00;

parity=0;

pkt_valid=1;

detect_addr=1;

header={payload_len,addr};

din=header;
```

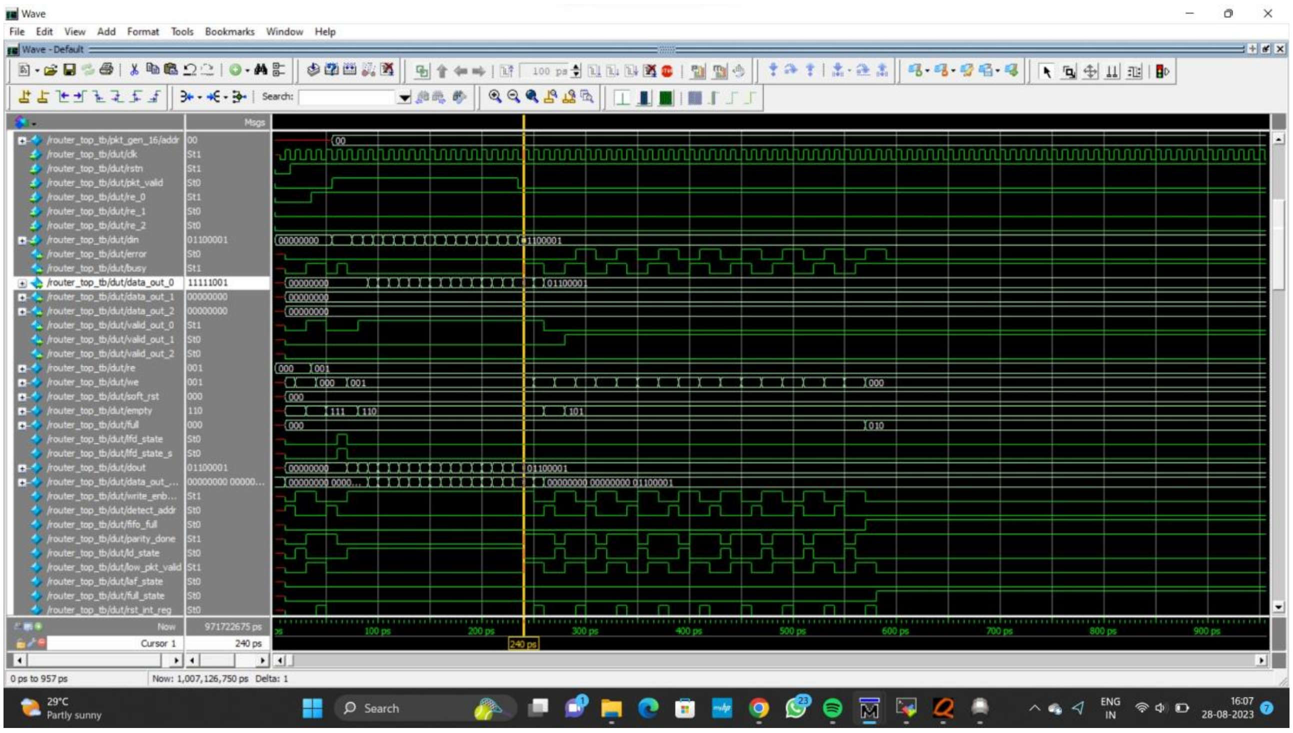
```
parity=parity^din;
@(negedge clk)
detect_addr=0;
lfd_state=1;
full_state=0;
fifo_full=0;
laf_state=0;
for(i=0;i<payload_len;i=i+1)
begin
@(negedge clk)
lfd_state=0;
ld_state=1;
payload_data={$random}%256;
din=payload_data;
parity=parity^din;
end
@(negedge clk)
pkt_valid=0;
din=parity;
@(negedge clk)
ld_state=0;
end
endtask
initial
begin
reset;
fifo_full=1'b0;
laf_state=1'b0;
full_state=1'b0;
#20;
pkt_gen();
#100;
$finish;
end
endmodule
```

```
module router_top_tb();
reg clk,rstn,pkt_valid;
reg re_0,re_1,re_2;
reg[7:0]din;
wire error,busy;
wire[7:0]data_out_0;
wire[7:0]data_out_1;
wire[7:0]data_out_2;
wire  valid_out_0,valid_out_1,valid_out_2;
parameter cycle=10;
integer i;
router_top
dut(clk,rstn,pkt_valid,re_0,re_1,re_2,din,data_out_0,data_out_1,data_out_2,valid_out_0,valid_out_1,valid_o
ut_2,error,busy);
always
begin
#(cycle/2);
clk=1'b0;
#(cycle/2);
clk=1'b1;
end
task rst_dut;
begin
@(negedge clk)
rstn=1'b0;
@(negedge clk)
rstn=1'b1;
end
endtask
task initialize;
begin
{re_0,re_1,re_2,pkt_valid,din}=0;
rstn=0;
end
endtask
task pkt_gen_16;
reg[7:0]payload_data;
reg[7:0]parity,header;
```

```
reg[5:0]payload_len;
reg[1:0]addr;
begin
@(negedge clk)
wait(~busy)
@(negedge clk)
payload_len=6'd16;
addr=2'b00;
header={payload_len,addr};
parity=0;
din=header;
pkt_valid=1;
parity=parity^header;
@(negedge clk)
wait(~busy)
for(i=0;i<payload_len;i=i+1)
begin
@(negedge clk)
wait(~busy)
payload_data={$random}%256;
din=payload_data;
parity=parity^din;
end
@(negedge clk)
wait(~busy)
pkt_valid=0;
din=parity;
end
endtask
initial
begin
initialize;
rst_dut;
#20;
fork
pkt_gen_16;
begin
```

```
wait(valid_out_0)
@(negedge clk);
re_0=1'b1;
#400;
end
join
end
endmodule
```

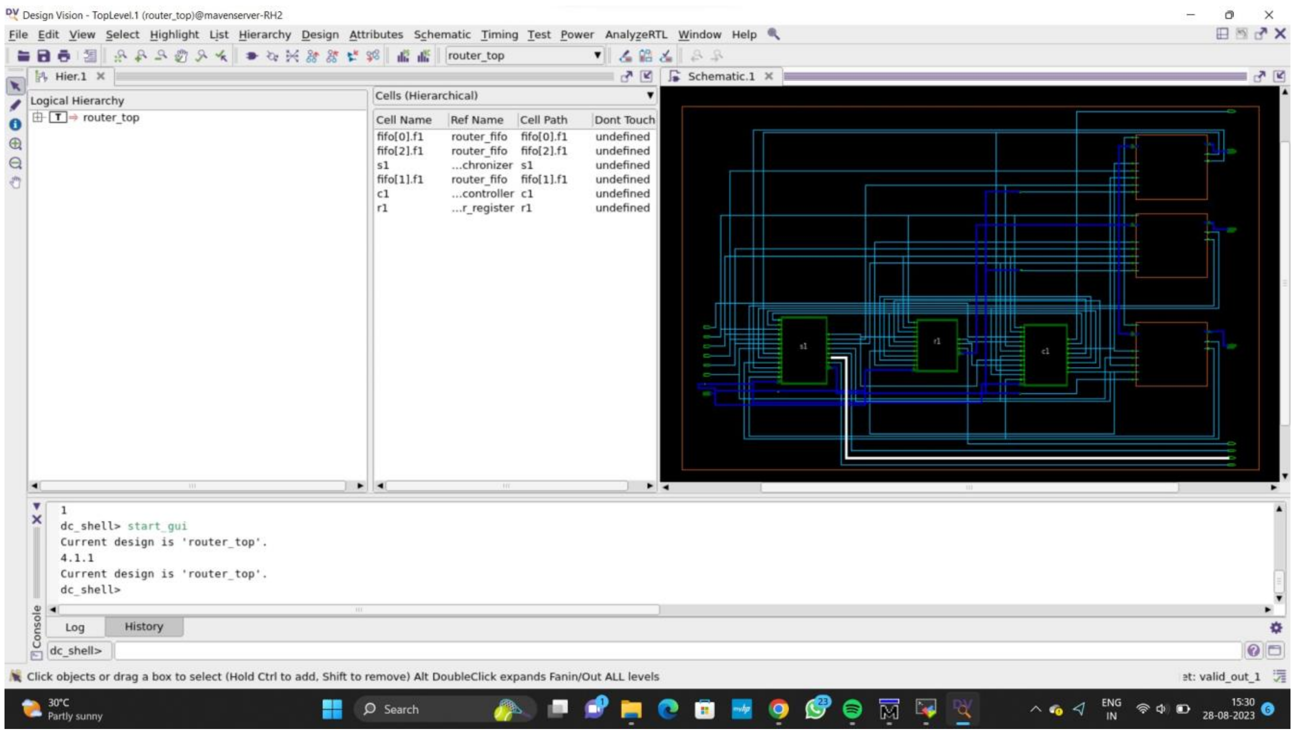
SIMULATION WAVEFORM:



	Msgs
/router_top_tb/pkt_gen_16/addr	00
/router_top_tb/dut/clk	St1
/router_top_tb/dut/rstn	St1
/router_top_tb/dut/pkt_valid	St0
/router_top_tb/dut/re_0	St1
/router_top_tb/dut/re_1	St0
/router_top_tb/dut/re_2	St0
/router_top_tb/dut/din	01100001
/router_top_tb/dut/error	St0
/router_top_tb/dut/busy	St1
/router_top_tb/dut/data_out_0	11111001
/router_top_tb/dut/data_out_1	00000000
/router_top_tb/dut/data_out_2	00000000
/router_top_tb/dut/valid_out_0	St1
/router_top_tb/dut/valid_out_1	St0
/router_top_tb/dut/valid_out_2	St0
/router_top_tb/dut/re	001
/router_top_tb/dut/we	001
/router_top_tb/dut/soft_rst	000
/router_top_tb/dut/empty	110
/router_top_tb/dut/full	000
/router_top_tb/dut/lfd_state	St0
/router_top_tb/dut/lfd_state_s	St0
/router_top_tb/dut/dout	01100001
/router_top_tb/dut/data_out_...	00000000 00000...
/router_top_tb/dut/write_enb...	St1
/router_top_tb/dut/detect_addr	St0
/router_top_tb/dut/fifo_full	St0
/router_top_tb/dut/parity_done	St1
/router_top_tb/dut/ld_state	St0
/router_top_tb/dut/low_pkt_valid	St1
/router_top_tb/dut/laf_state	St0
/router_top_tb/dut/full_state	St0
/router_top_tb/dut/rst_int_reg	St0
Now	971722675 ps
Cursor 1	240 ps

0 ps to 957 ps Now: 1,007,126,750 ps Delta: 1

SYNTHESIS REPORT:

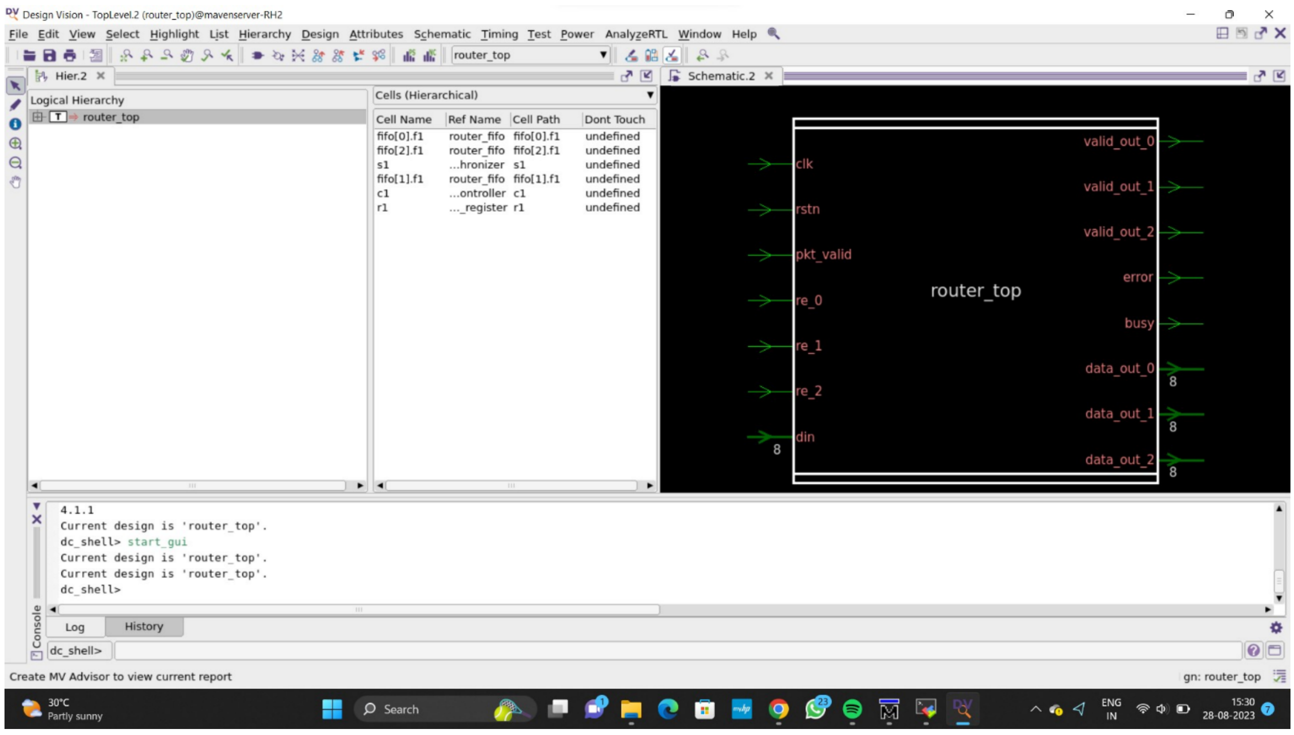

Design Vision - TopLevel.1 (router_top)@mavenserver-RH2
File Edit View Select Highlight List Hierarchy Design Attributes Schematic Timing Test Power AnalyzeRTL Window Help
router_top
Hier.1
Schematic.1
Logical Hierarchy
router_top
Cells (Hierarchical)
s1
r1
c1
Log History
Console
et: valid_out_1
30°C Partly sunny
Search
ENG IN
15:30 28-08-2023

Cell Name	Ref Name	Cell Path	Dont Touch
fifo[0].f1	router_fifo	fifo[0].f1	undefined
fifo[2].f1	router_fifo	fifo[2].f1	undefined
s1	...chronizer	s1	undefined
fifo[1].f1	router_fifo	fifo[1].f1	undefined
c1	...controller	c1	undefined
r1	...r_register	r1	undefined

```
1
dc_shell> start_gui
Current design is 'router_top'.
4.1.1
Current design is 'router_top'.
dc_shell>
```

dc_shell>

Click objects or drag a box to select (Hold Ctrl to add, Shift to remove) Alt DoubleClick expands Fanin/Out ALL levels



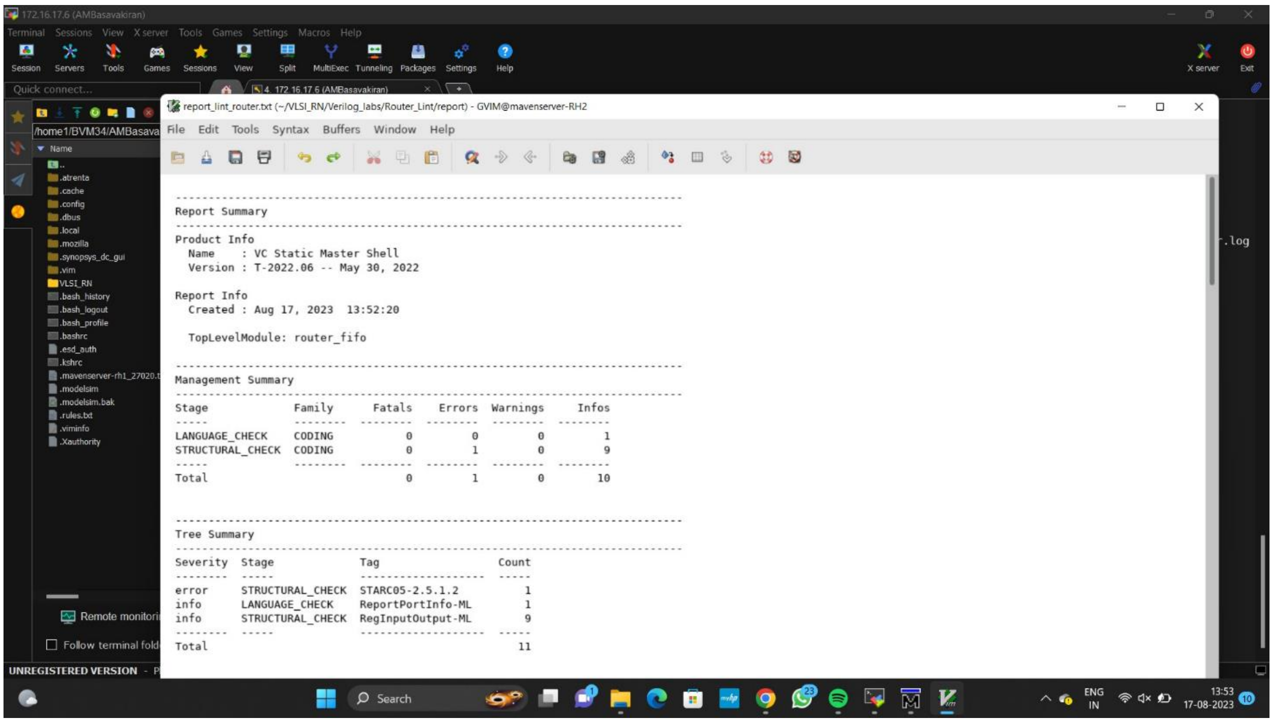
Cell Name	Ref Name	Cell Path	Dont Touch
fifo[0].f1	router_fifo	fifo[0].f1	undefined
fifo[2].f1	router_fifo	fifo[2].f1	undefined
s1	...hronizer	s1	undefined
fifo[1].f1	router_fifo	fifo[1].f1	undefined
c1	...ontroller	c1	undefined
r1	..._register	r1	undefined

```
4.1.1
Current design is 'router_top'.
dc_shell> start_gui
Current design is 'router_top'.
Current design is 'router_top'.
dc_shell>
```

dc_shell>

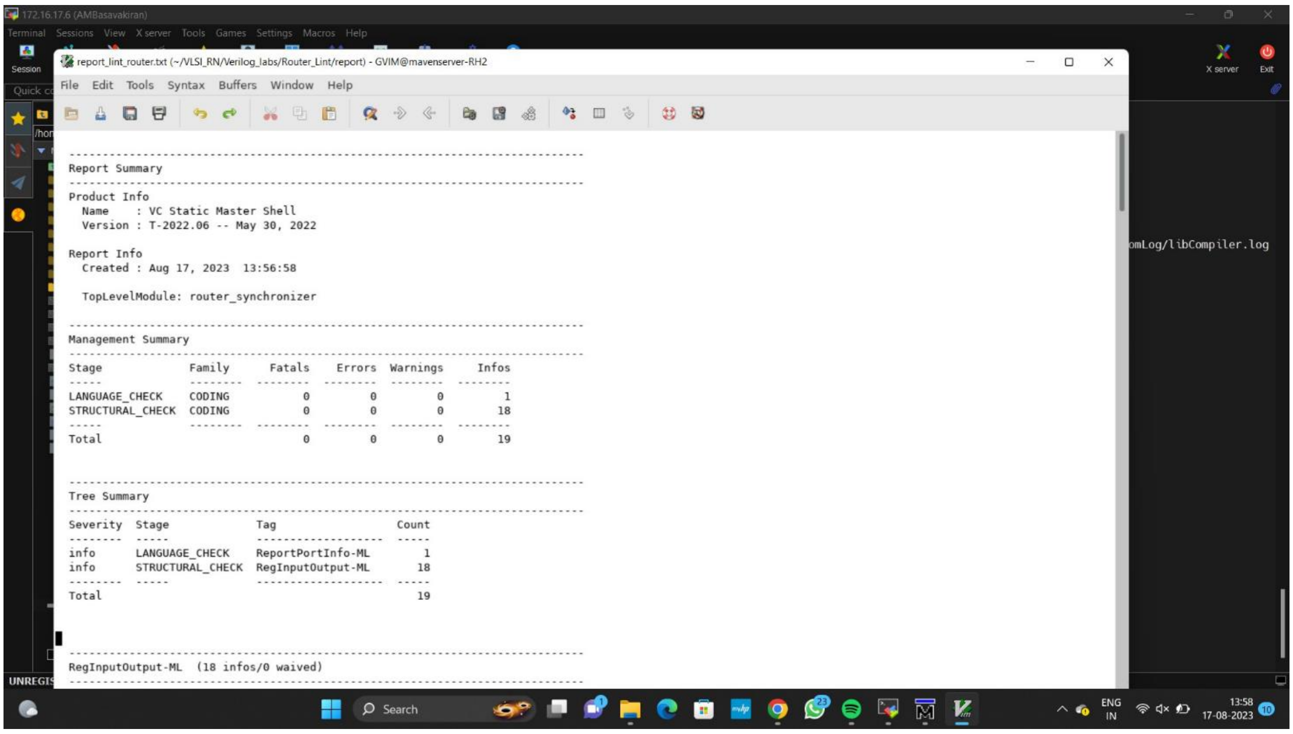
Create MV Advisor to view current report

ROUTER LINT REPORT:



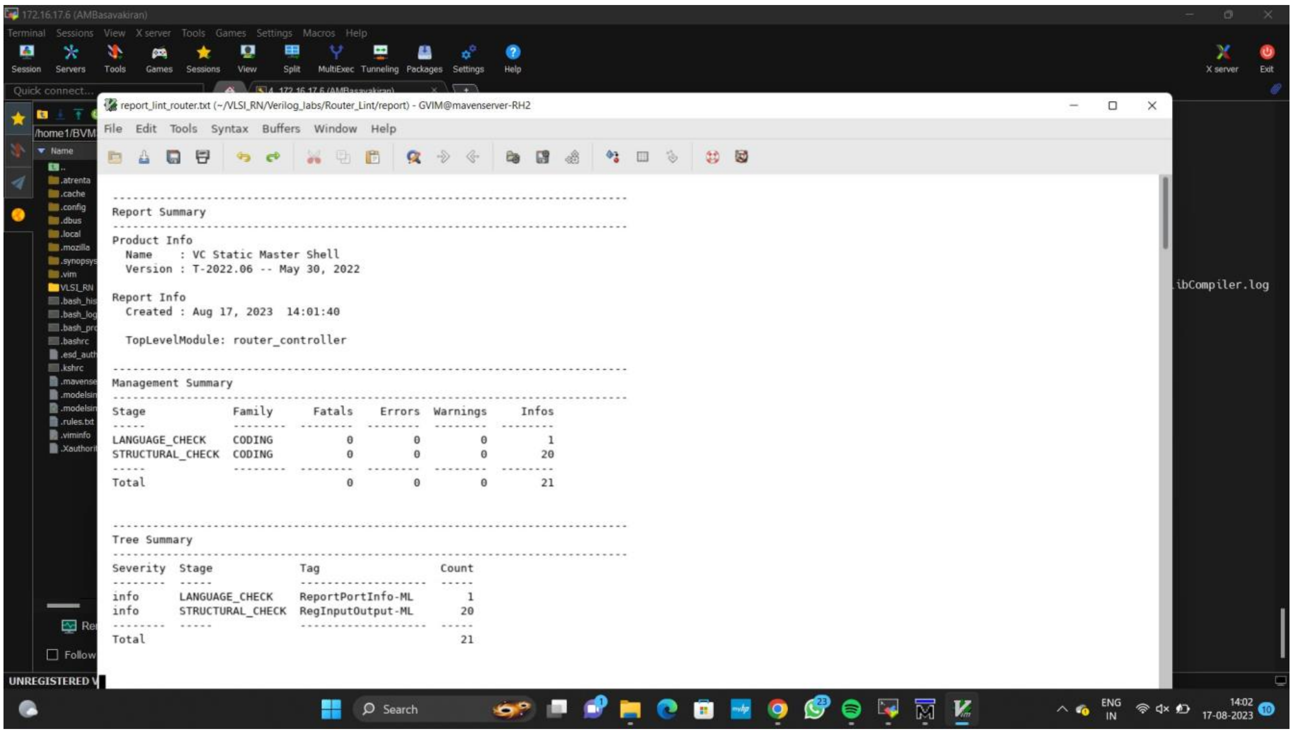
report_lint_router.txt (~/VLSI_RN/Verilog_labs/Router_Lint/report) - GVIM@mavenserver-RH2

```
---------------------------------------------------------------------------
Report Summary
---------------------------------------------------------------------------
Product Info
  Name    : VC Static Master Shell
  Version : T-2022.06 -- May 30, 2022

Report Info
  Created : Aug 17, 2023  13:52:20

  TopLevelModule: router_fifo

---------------------------------------------------------------------------
Management Summary
---------------------------------------------------------------------------
Stage            Family     Fatals   Errors  Warnings    Infos
-----            --------  --------  --------  --------  --------
LANGUAGE_CHECK   CODING         0        0        0        1
STRUCTURAL_CHECK CODING         0        1        0        9
-----            --------  --------  --------  --------  --------
Total                           0        1        0       10


---------------------------------------------------------------------------
Tree Summary
---------------------------------------------------------------------------
Severity  Stage            Tag                  Count
--------  -----            ------------------   -----
error     STRUCTURAL_CHECK STARC05-2.5.1.2          1
info      LANGUAGE_CHECK   ReportPortInfo-ML        1
info      STRUCTURAL_CHECK RegInputOutput-ML        9
--------  -----            ------------------   -----
Total                                              11
```

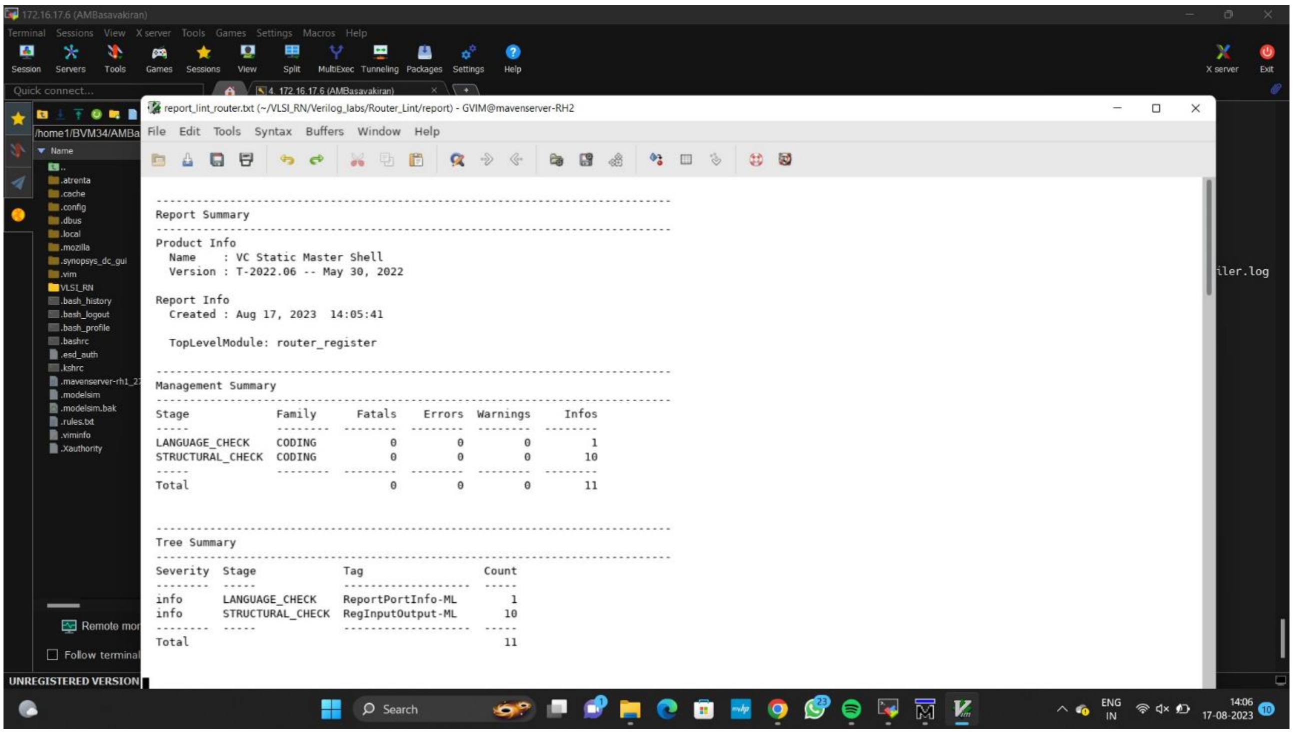


report_lint_router.txt (~/VLSI_RN/Verilog_labs/Router_Lint/report) - GVIM@mavenserver-RH2

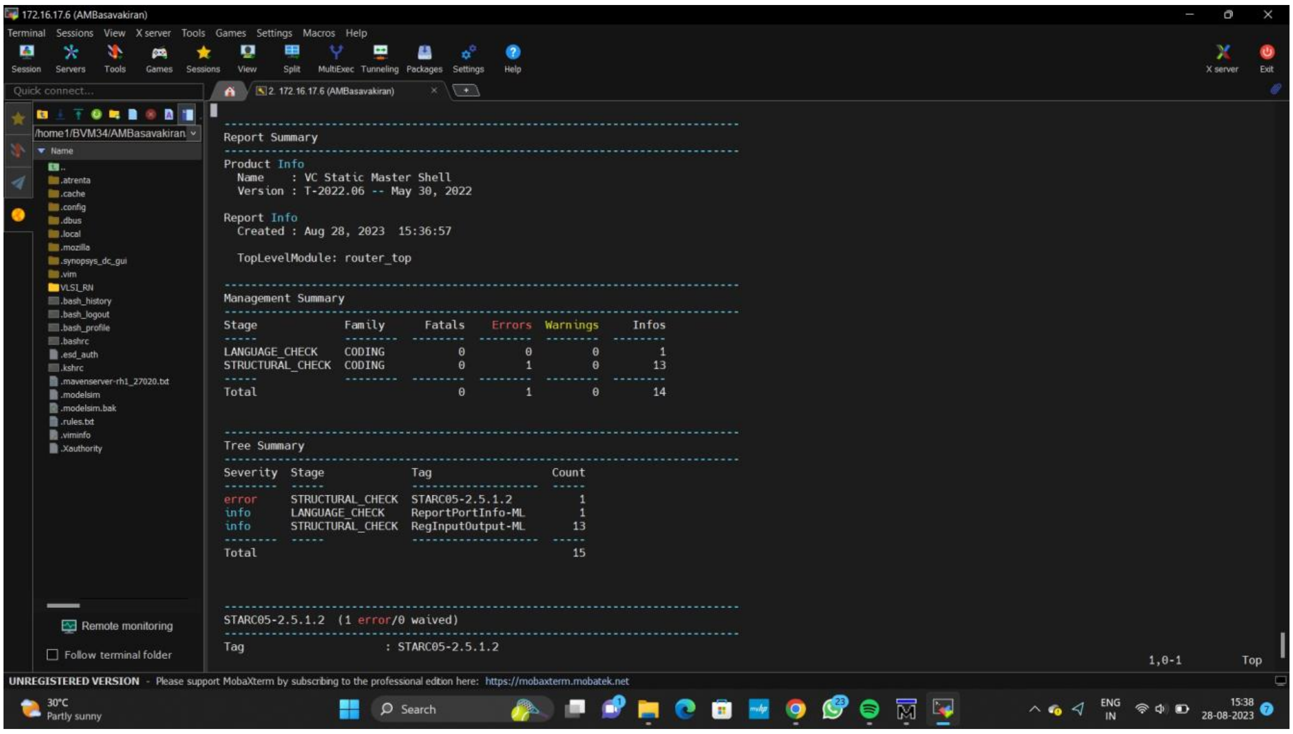
```
---------------------------------------------------------------------------
Report Summary
---------------------------------------------------------------------------
Product Info
  Name    : VC Static Master Shell
  Version : T-2022.06 -- May 30, 2022

Report Info
  Created : Aug 17, 2023  13:56:58

  TopLevelModule: router_synchronizer

---------------------------------------------------------------------------
Management Summary
---------------------------------------------------------------------------
Stage            Family     Fatals   Errors  Warnings    Infos
-----            --------  --------  --------  --------  --------
LANGUAGE_CHECK   CODING         0        0        0        1
STRUCTURAL_CHECK CODING         0        0        0       18
-----            --------  --------  --------  --------  --------
Total                           0        0        0       19


---------------------------------------------------------------------------
Tree Summary
---------------------------------------------------------------------------
Severity  Stage            Tag                  Count
--------  -----            ------------------   -----
info      LANGUAGE_CHECK   ReportPortInfo-ML        1
info      STRUCTURAL_CHECK RegInputOutput-ML       18
--------  -----            ------------------   -----
Total                                              19


---------------------------------------------------------------------------
RegInputOutput-ML  (18 infos/0 waived)
---------------------------------------------------------------------------
```



```
---------------------------------------------------------------------------
Report Summary
---------------------------------------------------------------------------
Product Info
  Name    : VC Static Master Shell
  Version : T-2022.06 -- May 30, 2022

Report Info
  Created : Aug 17, 2023  14:01:40

  TopLevelModule: router_controller

---------------------------------------------------------------------------
Management Summary
---------------------------------------------------------------------------
Stage            Family      Fatals    Errors  Warnings     Infos
-----            --------  --------  --------  --------  --------
LANGUAGE_CHECK   CODING           0         0         0         1
STRUCTURAL_CHECK CODING           0         0         0        20
-----            --------  --------  --------  --------  --------
Total                             0         0         0        21


---------------------------------------------------------------------------
Tree Summary
---------------------------------------------------------------------------
Severity  Stage            Tag                   Count
--------  -----            -------------------   -----
info      LANGUAGE_CHECK   ReportPortInfo-ML         1
info      STRUCTURAL_CHECK RegInputOutput-ML        20
--------  -----            -------------------   -----
Total                                               21
```



```
---------------------------------------------------------------------------
Report Summary
---------------------------------------------------------------------------
Product Info
  Name    : VC Static Master Shell
  Version : T-2022.06 -- May 30, 2022

Report Info
  Created : Aug 17, 2023  14:05:41

  TopLevelModule: router_register

---------------------------------------------------------------------------
Management Summary
---------------------------------------------------------------------------
Stage            Family      Fatals    Errors  Warnings     Infos
-----            --------  --------  --------  --------  --------
LANGUAGE_CHECK   CODING           0         0         0         1
STRUCTURAL_CHECK CODING           0         0         0        10
-----            --------  --------  --------  --------  --------
Total                             0         0         0        11


---------------------------------------------------------------------------
Tree Summary
---------------------------------------------------------------------------
Severity  Stage            Tag                   Count
--------  -----            -------------------   -----
info      LANGUAGE_CHECK   ReportPortInfo-ML         1
info      STRUCTURAL_CHECK RegInputOutput-ML        10
--------  -----            -------------------   -----
Total                                               11
```



```
------------------------------------------------------------------------
Report Summary
------------------------------------------------------------------------
Product Info
  Name    : VC Static Master Shell
  Version : T-2022.06 -- May 30, 2022

Report Info
  Created : Aug 28, 2023  15:36:57

  TopLevelModule: router_top

------------------------------------------------------------------------
Management Summary
------------------------------------------------------------------------
Stage             Family      Fatals    Errors  Warnings     Infos
-----             --------  --------  --------  --------  --------
LANGUAGE_CHECK    CODING           0         0         0         1
STRUCTURAL_CHECK  CODING           0         1         0        13
-----             --------  --------  --------  --------  --------
Total                              0         1         0        14


------------------------------------------------------------------------
Tree Summary
------------------------------------------------------------------------
Severity  Stage             Tag                   Count
--------  -----             ------------------    -----
error     STRUCTURAL_CHECK  STARC05-2.5.1.2           1
info      LANGUAGE_CHECK    ReportPortInfo-ML         1
info      STRUCTURAL_CHECK  RegInputOutput-ML        13
--------  -----             ------------------    -----
Total                                                15


------------------------------------------------------------------------
STARC05-2.5.1.2  (1 error/0 waived)
------------------------------------------------------------------------
Tag                    : STARC05-2.5.1.2
```

CHANGES DONE DURING ROUTER LINTING:

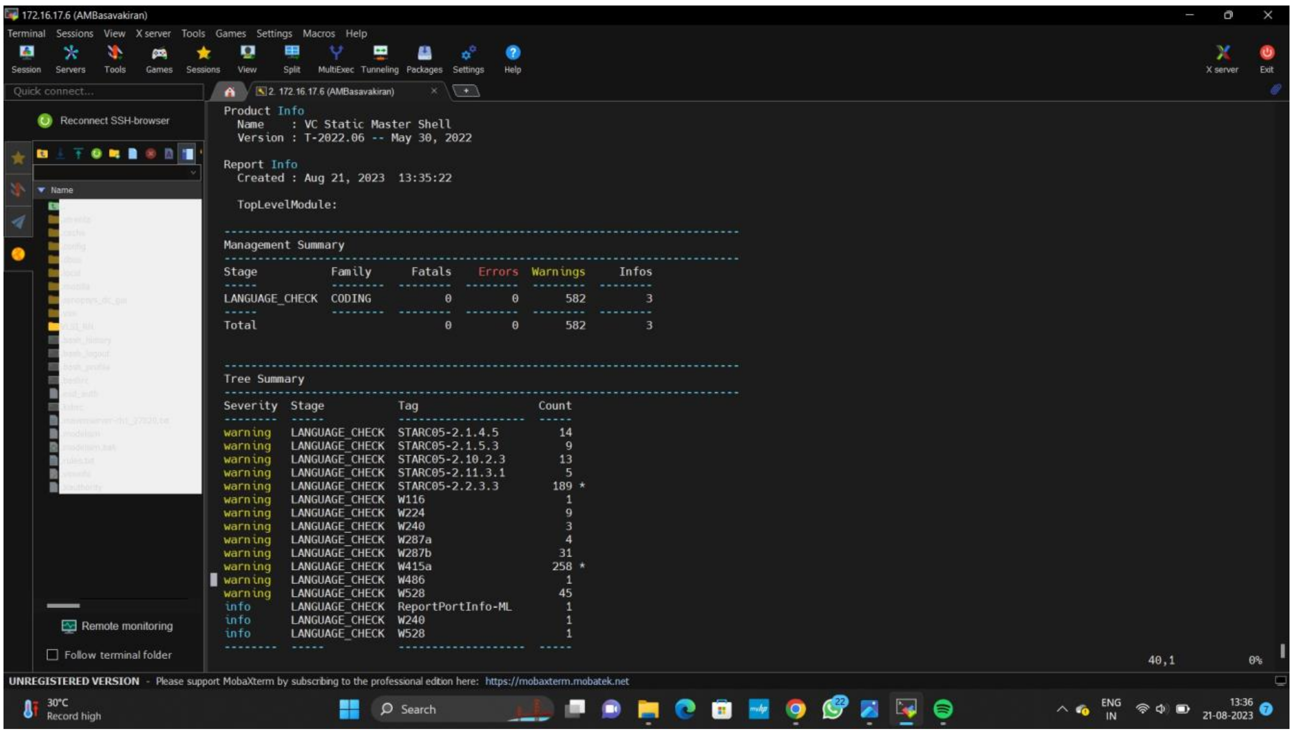
In Router_synchronizer.v file

```
wire w1=(timer_0==5'd29)?1'b1:1'b0;
wire w2=(timer_1==5'd29)?1'b1:1'b0;
wire w3=(timer_2==5'd29)?1'b1:1'b0;
always@(posedge clk)
begin
        if(~(rstn))
begin
                timer_0<=0;
        soft_rst_0<=0;
end
else if(valid_out_0)
begin
        if(!re_0)
        begin
                if(w1)
                begin
                        soft_rst_0<=1;
                        timer_0<=0;
                end
                else
                begin
```

```
                        soft_rst_0<=0;
                        timer_0<=timer_0+1'b1;
                end
        end
end
end
always@(posedge clk)
begin
        if(~(rstn))
begin
                timer_1<=0;
        soft_rst_1<=0;
end
else if(valid_out_1)
begin
        if(!re_1)
        begin
                if(w2)
                begin
                        soft_rst_1<=1;
                        timer_1<=0;
                end
                else
                begin
                        soft_rst_1<=0;
                        timer_1<=timer_1+1'b1;
                end
        end
end
end
always@(posedge clk)
begin
        if(~(rstn))
begin
                timer_2<=0;
        soft_rst_2<=0;
end
```

```
else if(valid_out_2)
begin
        if(!re_2)
        begin
                if(w3)
                begin
                        soft_rst_2<=1;
                        timer_2<=0;
                end
                else
                begin
                        soft_rst_2<=0;
                        timer_2<=timer_2+1'b1;
                end
        end
end
end
```

RISCV SOC LINT REPORT:


```
Product Info
  Name    : VC Static Master Shell
  Version : T-2022.06 -- May 30, 2022

Report Info
  Created : Aug 21, 2023  13:35:22

  TopLevelModule:

----------------------------------------------------------------------------
Management Summary
----------------------------------------------------------------------------
Stage           Family      Fatals    Errors  Warnings     Infos
-----           --------  --------  --------  --------  --------
LANGUAGE_CHECK  CODING           0         0       582         3
-----           --------  --------  --------  --------  --------
Total                            0         0       582         3


----------------------------------------------------------------------------
Tree Summary
----------------------------------------------------------------------------
Severity  Stage           Tag                   Count
--------  -----           -------------------   -----
warning   LANGUAGE_CHECK  STARC05-2.1.4.5          14
warning   LANGUAGE_CHECK  STARC05-2.1.5.3           9
warning   LANGUAGE_CHECK  STARC05-2.10.2.3         13
warning   LANGUAGE_CHECK  STARC05-2.11.3.1          5
warning   LANGUAGE_CHECK  STARC05-2.2.3.3         189 *
warning   LANGUAGE_CHECK  W116                      1
warning   LANGUAGE_CHECK  W224                      9
warning   LANGUAGE_CHECK  W240                      3
warning   LANGUAGE_CHECK  W287a                     4
warning   LANGUAGE_CHECK  W287b                    31
warning   LANGUAGE_CHECK  W415a                   258 *
warning   LANGUAGE_CHECK  W486                      1
warning   LANGUAGE_CHECK  W528                     45
info      LANGUAGE_CHECK  ReportPortInfo-ML         1
info      LANGUAGE_CHECK  W240                      1
info      LANGUAGE_CHECK  W528                      1
--------  -----           -------------------   -----
                                                                40,1          0%
```
172.16.17.6 (AMBasavakiran)
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
X server Exit
Quick connect...
2. 172.16.17.6 (AMBasavakiran)
Reconnect SSH-browser
Name
Remote monitoring
Follow terminal folder
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net
30°C Record high
Search
ENG IN
13:36 21-08-2023