FWC RTL ASSIGNMENT-2

Chirag Shah

Main Code

```
`timescale 1ns / 1ps
 module outer(
input clk
    );
    wire reset;
    wire [31:0] sum;
   vio_0 in6 (
  .clk(clk),                // input wire clk
  .probe_in0(sum) ,   // input wire [7 : 0] probe_in0
  .probe_out0(reset) // output wire [0 : 0] probe_out0
);
top_module(.clk_100M(clk),.reset(reset),.sum(sum));
endmodule

//top module
`timescale 1ns / 1ps
module outer(
input clk
    );
    wire reset;
    wire [31:0] sum;
   vio_0 in6 (
  .clk(clk),                // input wire clk
```

```
  .probe_in0(sum) ,   // input wire [7 : 0] probe_in0
  .probe_out0(reset) // output wire [0 : 0] probe_out0
);
top_module(.clk_100M(clk),.reset(reset),.sum(sum));
endmodule

//block memory generator with ILA block
`timescale 1ns / 1ps
module bmg_fib(
input clk,
    input clk2,
    input rst,
    output [31:0] sum
    );
reg [31:0] addr_next;
wire [31:0] douta;
wire [31:0] doutb;
reg [31:0] dina=0;
reg [31:0] dinb=1;
reg wea=1;
reg web=1;
reg [6 : 0] addra=0;
reg [6 : 0] addrb=1;
//reg [31:0] ram[63:0];

//address will increment  in this loop
always@(posedge clk)
```

```
begin
addra<=addr_next;
addrb<=addr_next+1;
end
always@(*)
begin
addr_next=0;
else
addr_next=addra+1;
end

  blk_mem_gen_0 in4 (
 .clka(clka),    // input wire clka
 .wea(wea),      // input wire [0 : 0] wea
 .addra(addra),  // input wire [6 : 0] addra
 .dina(dina),    // input wire [31 : 0] dina
 .douta(douta),  // output wire [31 : 0] douta
 .clkb(clkb),    // input wire clkb
 .web(web),      // input wire [0 : 0] web
 .addrb(addrb),  // input wire [6 : 0] addrb
 .dinb(dinb),    // input wire [31 : 0] dinb
 .doutb(doutb)  // output wire [31 : 0] doutb
);

//incrementing the fib
always @ (posedge clk)
begin
```

```
if (rst==1'b1)
begin
dina<= 1'b0; //PREVIOUS VALUE
dinb<= 1'b1; //CURRENT VALUE
end
else if(clk==1)
begin
dina<=dinb;
dinb<=dina+dinb;
end
end
assign sum=dina;

ila_0 in5 (
.clk(clk2), // input wire clk

.probe0(clk),
.probe1(addra), // input wire [0:0]  probe0
.probe2(sum), // input wire [5:0]  probe1
.probe3(rst) // input wire [31:0]  probe2
);
endmodule
```

Test bench

```
//for Block memory generator
`timescale 1ns / 1ps
module bmg_fib_tb();
   reg clk;
```

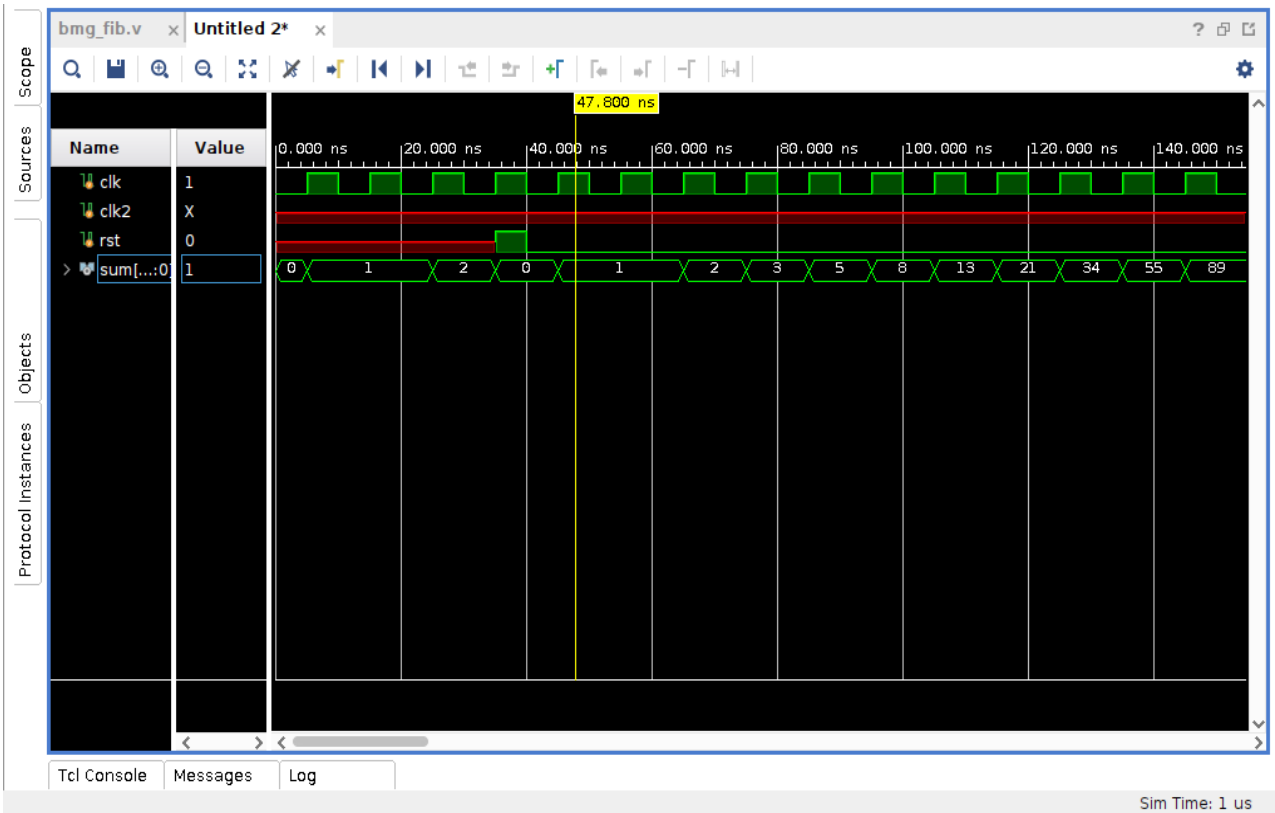
```
    reg clk2;
    reg rst;
    wire [31:0] sum;

bmg_fib tb3 (.clk(clk),.clk2(clk2),.rst(rst),.sum(sum));
initial
clk=0;
     always #5 clk=~clk;
   initial
   begin
   #35 rst=1;
   #5 rst=0;
   end
endmodule

//for converting 5MH CLK to 1H
`timescale 1ns / 1ps
module clk_div_tb();
    reg clk_in;
    wire divided_clk ;
    clk_divider tb1 (.clk_in(clk_in),.divided_clk(divided_clk));
    initial
    clk_in = 0;
        always #5 clk_in=~clk_in;
endmodule

//For converting 100MH CLK to 5MH CLK
`timescale 1ns / 1ps
module clk_wiz_tb();
```

```
    reg clk_100MH;
    wire clk_5MH ;
    top_module tb1 (.clk_100M(clk_100MH),.clk_5M(clk_5MH));
    initial
    clk_100MH = 0;
       always #5 clk_100MH=~clk_100MH;
endmodule
```

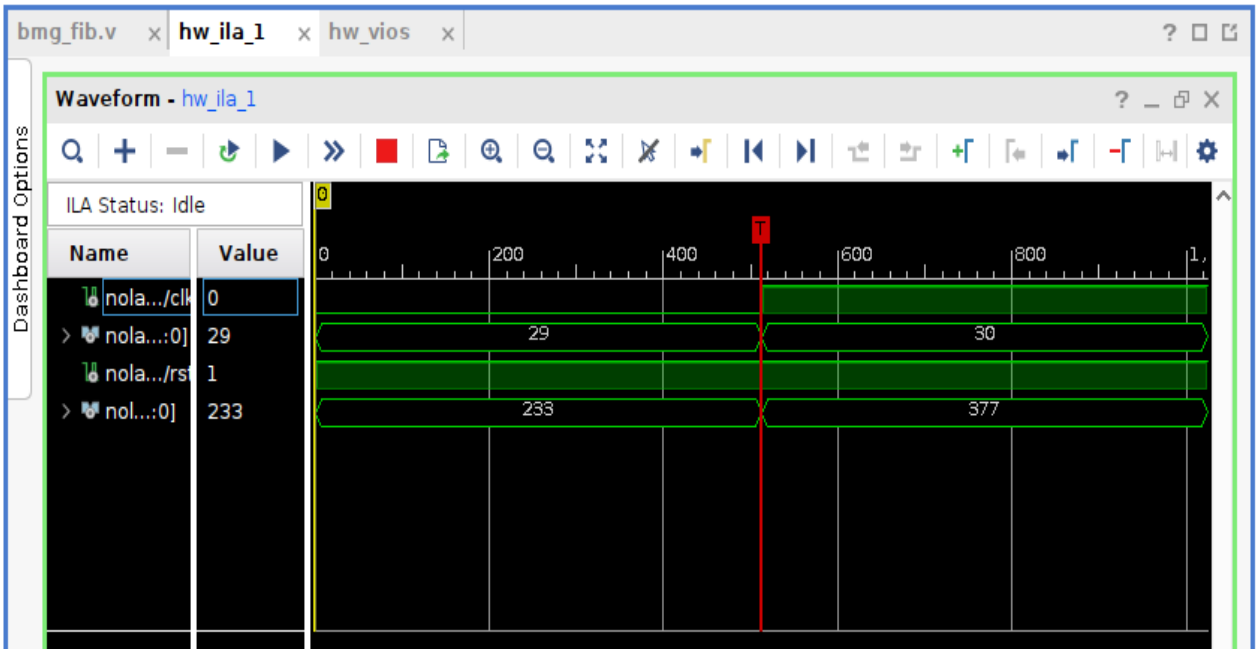
Constraints

```
set_property -dict{PACKAGE_PIN E3 IOSTANDARD LVCMOS18 }[get_ports {clk}];
create_clock -add -name sys_clk_pin -period 8.00 -waveform{0 4}[get_ports {clk}];
```

OVERALL SIMULATION



WORKING (ILA)



Clk Simulation (100MH to 5MH)