INDIAN INSTITUTE OF TECHNOLOGY PALAKKAD

EE5011 VLSI Design Lab Report: Experiment 3

Karthik Dhondi
IIT Palakkad/M.Tech System On Chip Design
Email: 152202007@smail.iitpkd.ac.in

Abstract—Experiment 3 is to Generate the N_{th} Fibonacci number which takes 3-bits N-value as input.It has two qualifiers, $N-valid$ and $Fibo-valid$,where $N-valid$ is to validate the N-value and whenever the $Fibo-valid$ is high then it should output the N_{th} fibonacci of the given input N-Value.

I Introduction

In this, For a given Fibonacci series we are generating the N_{th} Fibonacci number.By the given two qualifiers they are N_{valid} and $Fibo_{valid}$ where an input N is taken only when the N_{valid} is high, it will take the input and it shows the output whenever the $Fibo_{valid}$ is high.Not only single input but also it will take multiple N values as input and it shows all the outputs in a given order of inputs, whenever the $Fibo_{valid}$ is high.

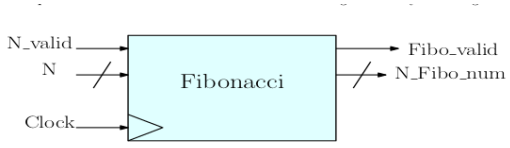

Fig. 1: Block Diagram

II Implementation Details

It is same as implementing the Fibonacci series generator, as it will takes two previous numbers and add it and gets the new value and it is done in recursive manner. The main challenge in this experiment is we are giving some random $N-value$, that value is accepted only when the $N-valid$ is high and it will show the exact N_{th} Fibonacci number when the $Fibo-valid$ is high. Then the $N-Fibo-num$ clock pulse will be high for the respective input. For the above function we have done a verilog code, which implements the exact functionality as described. The fig.2 shows the verilog code implementation.

For the above verilog code a simple testbench developed in Verilog to test the DUT. The respective test bench is shown below in fig.3

```
module fib_n(
  input clk,
  input reset,
  input N_valid,
  input [2:0] N,
  output reg [15:0] Q1,
  output reg Fibo_valid,
  output reg [15:0] Fibo_out
);

reg  [15:0] Q0;
wire [15:0] D0;
reg [2:0] count;

assign D0 = Q0+Q1;

always @(posedge clk) begin

    if (N_valid) begin Q0 <= 1;  Q1 <= 0;  count <= 0; Fibo_valid <= 0; end
    else begin         Q0 <= D0; Q1 <= Q0; count <= count + 1; end

    if (count == N) begin  Fibo_valid <= 1;   Fibo_out <= Q1;  end
    else   begin           Fibo_valid <= 0;   Fibo_out <= 0;   end

end
endmodule
```

Fig. 2: Verilog Code for N_{th} Fibonacci number

```
module fibonacci_tb;
reg clk;
reg reset;
wire [15:0] Q1;
wire [15:0] Fibo_out;
wire Fibo_valid;
reg N_valid;
reg [2:0] N;


fib_n DUT(.clk(clk), .reset(reset),.N(N),.N_valid(N_valid),
          .Q1(Q1),.Fibo_valid(Fibo_valid),.Fibo_out(Fibo_out));

always #2 clk = ~clk;

initial
    begin
        $dumpfile("dump.vcd");
        $dumpvars(0);

        clk <= 0;N_valid<= 0;
        #2;

        N <= 3'b 110;
        #2;
        N_valid <= 1;
        #2;
        N_valid <= 0;

        #40;

        N <= 3'b 101;
        #2;
        N_valid <= 1;
        #2;
        N_valid <= 0;


        #100;

        $finish;
    end

endmodule
```

Fig. 3: Verilog Testbench for N_{th} Fibonacci number

III Physical Design

Once we are done with the synthesis,we do Physical Design, using **INNOVUS tool**. To do the physical design, we

need to import the design that is we are getting the netlist file. Next we are setting the Power constraints to VDD as Power nets and VSS as Ground nets. Later we are doing the **Floor plan**, i.e, we are specifying the aspect ratio, core utilization. Next we are setting up the power by adding ring, in this specific design we took Metal9(9)H as horizontal and Metal8(8)V as vertical Ring configuration.Next in power we add stripes. Atlast, we do placement and timing.We get the Post CTS timing reports.

IV Results

The Verilog description of the N_{th} Fibonacci number is verified through simulation using Cadance as the simulation tool. As expected, the implementation produced a $N-Fibo-valid$ whenever the $Fibo-valid$ is high.The respective result can be seen below in fig.4

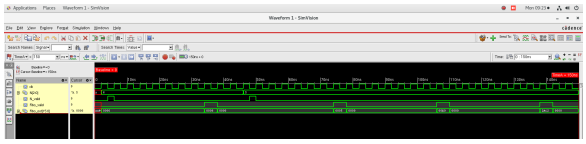

Fig. 4: Output Wave Form

Synthesis of the above implementation is done using Genus tool,schematic of the N_{th} Fibonaaci shown in fig.5

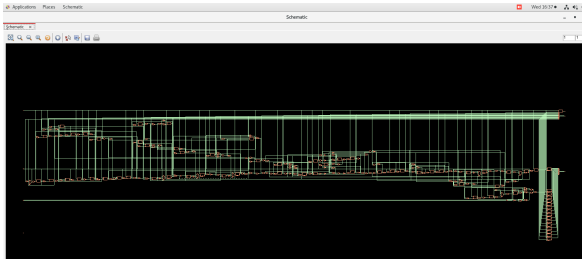

Fig. 5: Schematic of N_{th} Fibonacci number

The Table.1 shows the power consumption of the circuit N_{th} Fibonacci

Table 1: Power

Sl.no	Instance	Cells	Leakage Power(nW)	Dynamic Power(nW)	Total Power(nW)
1	fib_n	114	7784.022	170395.440	178179.463

The Table 2 shows the area consumed by the sequential ,

Table 2: Area

Sl.no	Type	Instances	Area	Area Perct.
1	sequential	52	899.197	64.2
2	logic	62	501.068	35.8
3	physical-cells	0	0.000	0.000
	Total	114	1400.265	100.0

In the timing the **Timing slack is 560ps**

$$Start-point : Q0_reg[0]/CK$$

$$End-point : Q0_reg[14]/D$$

The above start-point to end-point is the **critical path.**

V Results Of Physical Design

The below fig.6 is the Physical design of the N_{th} Fibonacci.The setup and hold times of this design are positive as we can see the

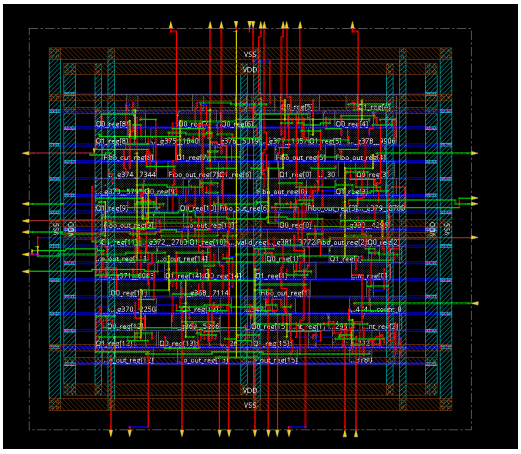

Fig. 6: Physical Design of N_{th} Fibonacci number

As we see the timing reports of Pre CTS and Post CTS

Table 3: Pre CTS Setup

Sl.no	Setup mode	all	reg2reg	default
1	WNS (ns)	5.443	5.443	8.701
2	TNS (ns)	0.000	0.000	0.000
3	Violating Paths	0	0	0
4	All Paths	68	68	52

Density: 69.654 Percent
Routing Overflow: 0.00 Percent H and 0.00 Percent V

Table 4: Post CTS Setup

Sl.no	Setup mode	all	reg2reg	default
1	WNS (ns)	5.450	5.450	8.978
2	TNS (ns)	0.000	0.000	0.000
3	Violating Paths	0	0	0
4	All Paths	68	68	52

Density: 75.264 Percent
Routing Overflow: 0.00 Percent H and 0.00 Percent V

As we see, in the Timing reports of the Pre CTS summary and Post CTS summary there is no much difference.

The Below is the Post CTS Hold summary.

Density: 75.264 pecntage
Routing Overflow: 0.00 Percent H and 0.00 Percent V

TABLE 5: POST CTS HOLD

Sl.no	**Hold mode**	**all**	**reg2reg**	**default**
1	WNS (ns)	0.002	0.002	0.005
2	TNS (ns)	0.000	0.000	0.000
3	Violating Paths	0	0	0
4	All Paths	68	68	52

VI CONCLUSION

The N_{th} Fibonacci generator is successfully implemented in Verilog HDL and verified through simulation using Cadance simvison tool, done synthesis through Cadance Genus tool and also done the Physical Design using the INNOVUS tool.After doing all the Experiment, we got Positive time slack of the Pre CTS steup time, Post CTS setup time and Post Cts of the Hold time. Postive time slack makes sure that, our expected output of the design is correct.Hence, N_{th} Fibonacci generator is designed perfectly.