Experiment #2 - Frequency Regulation

Mohammad.heydari
810197494

INTRODUCTION

In this experiment we try to make a block to eliminate the noise from the output signals.as we know each digital signals may be consist noises and this event may have destructive effects , then in this experiment for the special case we try to design an adjustable clock generator that can fix the output frequency at a desired value. In addition to we learn a bit about issues such as The concept of adjusting clock frequency, Hardware design & Writing test-bench and simulation and etc.

1 DESIGN DESCRIPTION AND SYNTHESIS

According to the describtion of lab we know for this case we need a pre processing unit to handle optimal frequency range.

1.1 Pre-Processing unit

The pre-processing unit called “Freq-Range-Spec” is a simple logic block that receives two input values ”fset-max” and ”fset-min” as parameters. The user also provides a reference regulating value ”fset-ref” as the input of this block. The reference value is compared with maximum and minimum boundaries and a decision is made on the final frequency setting value ”Setperiod”.

1. Write a Verilog code for this part and include it as a separate module in your design.

<u>The verlog code of this part have been attached with all of details.</u>

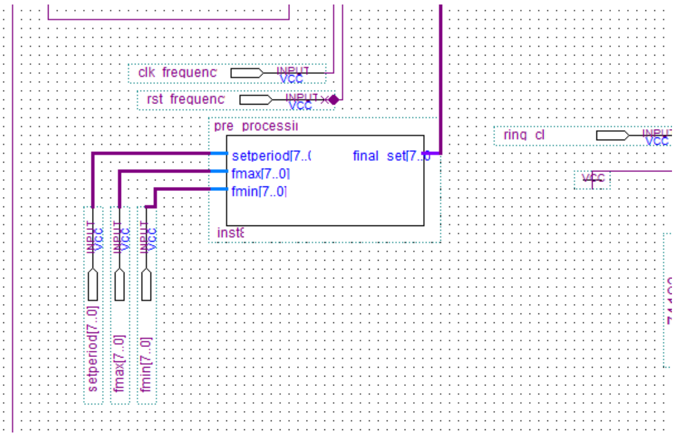


Fig. 1 separate Pre_processing unit in my design

2. Draw the hardware of this module (on the paper) using RTL components.

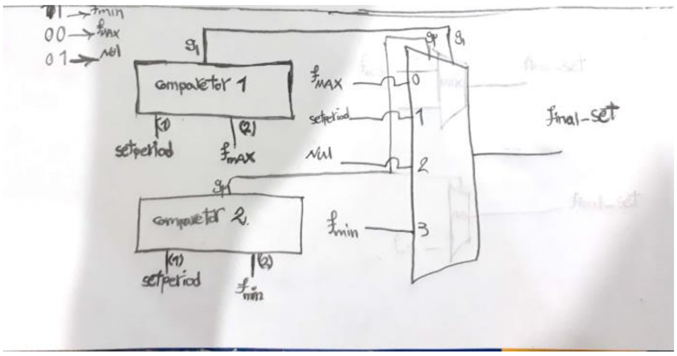


Fig. 2 hardware of preprocess unit

1.2 Main Processing unit

in this part we write a Verilog code for the main processing unit called Frequency-Regulator.

The regulation is performed by changing the value of the counter loads in clock divider. The adjusted value for division (counter load values) after processing is called adjustedDiv. A reference value called setPeriod is also another input that represents the desired output frequency in terms of number of clock cycles in one time duration.

The regulation is performed by changing the value of the counter loads in clock divider. The adjusted value for division (counter load values) after processing is called adjustedDiv. A reference value called setPeriod is also another input that represents the desired output frequency in terms of number of clock cycles in one time duration.

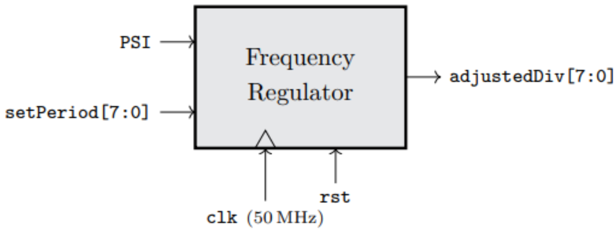


Fig. 3 Frequency regulator pins

According to above discribtion we are going to complete three always based on desired details.

Questions:

1. Write a Verilog code based on description above.

2. Synthesize this code as a top-level entity.

3. Create a symbol for this frequency regulator module.

4. Add the pre-processing block of section 1.1 to this block diagram and set it as the top-level entity.

5. set the necessary inputs and outputs. Note that the Frequency Regulator module needs a 50 MHz clock signal that is separate from the divider clock.

6. Synthesize the top-level design.

7. Include all the synthesis results, the Quartus II files in your report.

Cuase of relation of the questions, they have been answered together.

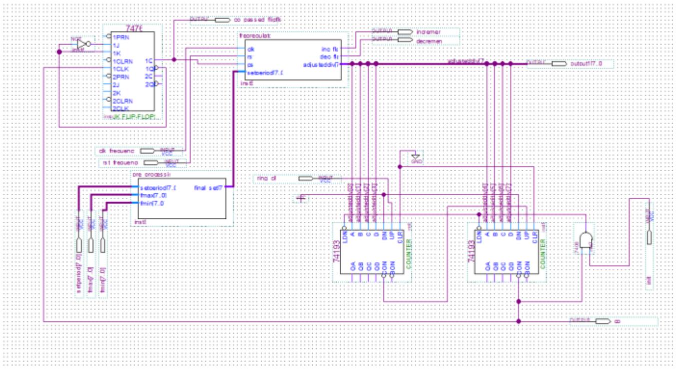


Fig. 4 schematic design in qurtus

In this part I write a verilogcode base on three always to adjust the loads. you can observe the details in my codes.after that I create the symbols of pre_process and frequency requlator and then I start wiring with two counter from the previous experiment. And the end part I Synthesize the top-level design and get .vo & .sdo output.

all the synthesis results, the Quartus II files have been attached

1.3 Clock divider unit

1. Add the clock division unit of Experiment 1 to the block diagram of section 1.2.

2. Make the necessary connections.

3. Synthesize this block as a top-level entity.

4. Include all the synthesis results, the Quartus II files in your report.

All of the requirement for this part already done in previuos part.

But I add a discribtion about that . we transfer our design from the previous experiment to this schematic and only . make some of connections between the old blocks & the new blocks such as Pre_prosecc & frequency_regulator.

all the synthesis results, the Quartus II files have been attached

2 Design Simulation in Modelsim

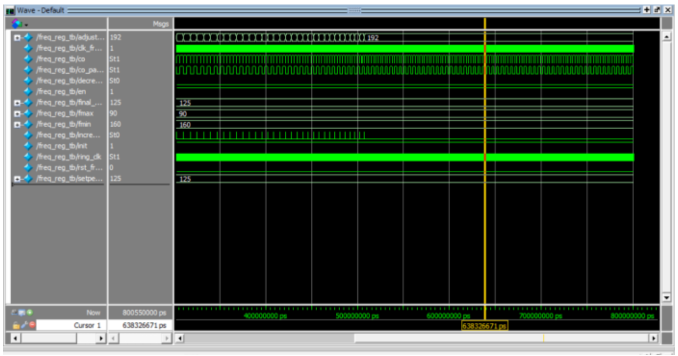
After synthesizing our design, we prepared to verify the hardwer Then we have following steps.

1. in first step we write a testbench according to general knowledge.
Note: I use the ring oscillator with a frequency smaller than 50 MHz (near 25 MHz).

2.
In this part we use following values :
Fmin=160 , fmax=90 , setperiod=125 , desired frequency=400k , initial = 127

Then 25M/400k=62.5 , 255-final_load=62.5

So we have final_load=255-63 = 192
And the result is completely the same ☺
As you see in following figure .

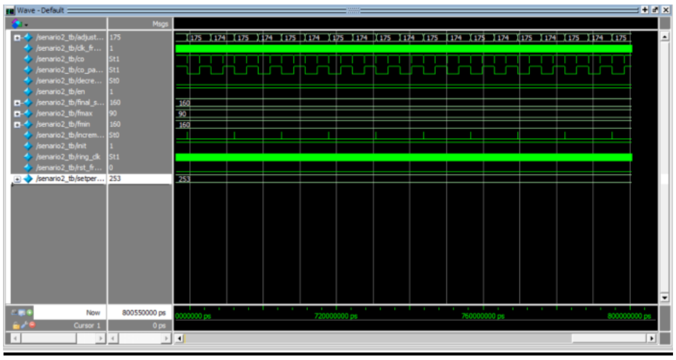


```
Wave - Default
                              Msgs
/freq_reg_tb/adjust...        192
/freq_reg_tb/clk_fr...        1
/freq_reg_tb/co               St1
/freq_reg_tb/co_pa...         St1
/freq_reg_tb/decre...         St0
/freq_reg_tb/en               1
/freq_reg_tb/final_...        125
/freq_reg_tb/fmax             90
/freq_reg_tb/fmin             160
/freq_reg_tb/incre...         St0
/freq_reg_tb/init             1
/freq_reg_tb/ring_clk         St1
/freq_reg_tb/rst_fr...        0
/freq_reg_tb/setpe...         125

Now        800550000 ps
Cursor 1   638326671 ps
```

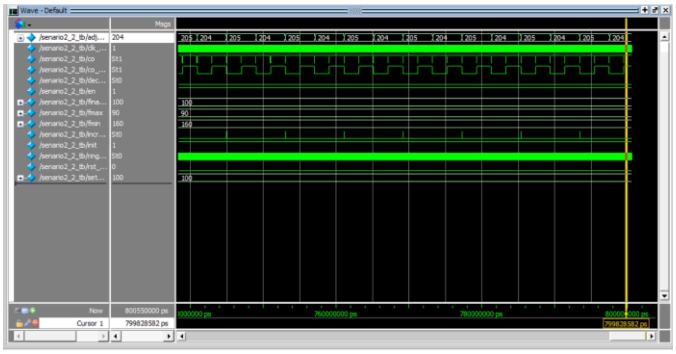
Fig. 5 results of scenario 1

3.



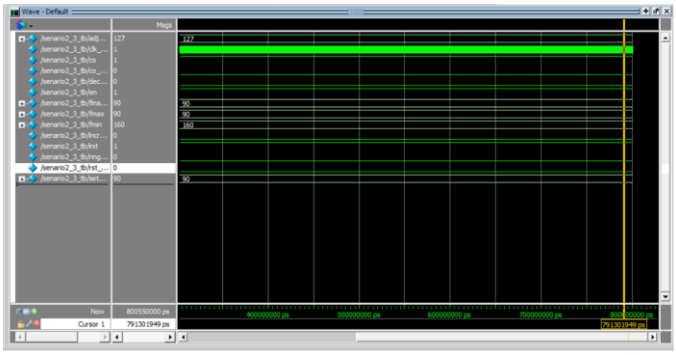
	Msgs
/senario2_tb/adjust...	175
/senario2_tb/clk_fr...	1
/senario2_tb/co	St1
/senario2_tb/co_pa...	St1
/senario2_tb/decre...	St0
/senario2_tb/en	1
/senario2_tb/final_s...	160
/senario2_tb/fmax	90
/senario2_tb/fmin	160
/senario2_tb/increm...	St0
/senario2_tb/init	1
/senario2_tb/ring_clk	St1
/senario2_tb/rst_fr...	0
/senario2_tb/setper...	253
Now	800550000 ps
Cursor 1	0 ps

Fig. 6 results of scenario 2 out of range_upper



	Msgs
/senario2_2_tb/adj...	204
/senario2_2_tb/clk_...	1
/senario2_2_tb/co	St1
/senario2_2_tb/co_...	St1
/senario2_2_tb/dec...	St0
/senario2_2_tb/en	1
/senario2_2_tb/fina...	100
/senario2_2_tb/fmax	90
/senario2_2_tb/fmin	160
/senario2_2_tb/incr...	St0
/senario2_2_tb/init	1
/senario2_2_tb/ring...	St0
/senario2_2_tb/rst_...	0
/senario2_2_tb/set...	100
Now	800550000 ps
Cursor 1	799828582 ps

Fig. 7 results of scenario 2 in range



	Msgs
/senario2_3_tb/adj...	127
/senario2_3_tb/clk_...	1
/senario2_3_tb/co	1
/senario2_3_tb/co_...	0
/senario2_3_tb/dec...	0
/senario2_3_tb/en	1
/senario2_3_tb/fina...	90
/senario2_3_tb/fmax	90
/senario2_3_tb/fmin	160
/senario2_3_tb/incr...	0
/senario2_3_tb/init	1
/senario2_3_tb/ring...	0
/senario2_3_tb/rst_...	0
/senario2_3_tb/set...	90
Now	800550000 ps
Cursor 1	791301949 ps

Fig. 8 results of scenario 2 out of range_lower

As you see in above figures those include the modelsim waveforms for these signals: duration, increment, decrement, setperiod, Ring oscillator clock, 50 MHz clock, PSI, and adjusteddiv according to the lab_discribtion.

In scenario 1 we set the value of frequency and setperiod and then try to simulate the noise of frequency in input with changing the frequency of ring scillator and then we see that the desired frequency catched by setperiod of 125 as mentioned in discribtion of Lab.

In scenario 2 we are going to see the task of pre process in which case that we gives setperiod three values that included both values within 160 and 90 and values outside this range the result have been attached.

4.

All of the results includes cursors & the output result of both quartus & modlesim have been attached.

Conclusions

IN THIS EXPERIMENT WE ARE GOING TO DESIGN AN ADJUSTABLE CLOCK GENERATOR THAT CAN FIX THE OUTPUT FREQUENCY AT A DESIRED VALUE AS WE KNOW ALWAYS IN ANY CONDITION WE FACED WITH MANY NOICE SIGNAL ON THE FREQUENCY OF RING OSCILLATOR THEN IN THIS EXPERIMENR WE DESIGN AN ADJUSTABLE CLOCK GENERATOR TO FIX THIS ISSUE.

References

[1] Some entries at github

[2] Some entries at stackoverflow.com

[3] Some entries to debug my codes