Experiment #2 – Frequency Regulation

Daneshvar Amrollahi
Student ID: 810197685

***Abstract*— This document is a student report to experiment #2 of Digital Logic Laboratory course at ECE Department, University of Tehran. In the previous experiment we implemented some clock generation methods such as Ring Oscillator, Schmitt Trigger Oscillator, and etc. The goal of this experiment is to design an adjustable clock generator that can fix the output frequency at a desired value.**

***Keywords*— Clock, Ring Oscillator, Frequency Regulation, Synchronization, Cyclone IV E**

I. Introduction

Synchronization is a crucial issue in sequential digital circuit design. An overall view of the adjustment system is showed in figure 1. It consists of 3 main units:

1. Freq-Range-Spec
2. Frequency-Regulator (main processing unit)
3. Clock Divider (Experiment #1)

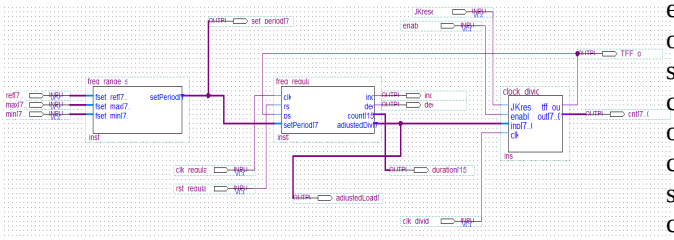

Fig. 1 Overall view of clock adjuster

II. FREQ-RANGE-SPEC UNIT

This part of the adjuster is a combinational unit. It has 3 inputs:

1. fset_ref[7:0]
2. fset_max[7:0]
3. fset_min[7:0]

It guarantees that fset_ref[7:0] will be in the range (fset_min[7:0] , fset_max[7:0]). If it is larger than fset_max[7:0], then fset_max[7:0] would be set for setPeriod[7:0]. If it is less than fset_min[7:0], then fset_min[7:0] would be set for setPeriod[7:0]. Otherwise, fset_ref[7:0] itself would be set for setPeriod[7:0]. setPeriod[7:0] is passed to the main processing unit which we will discuss in the next section of this report.

Figure 2 illustrates an RTL design for this unit using comparators and multiplexers. The comparator outputs a 1/0 signal indicating if the first input is larger than the second input or not.

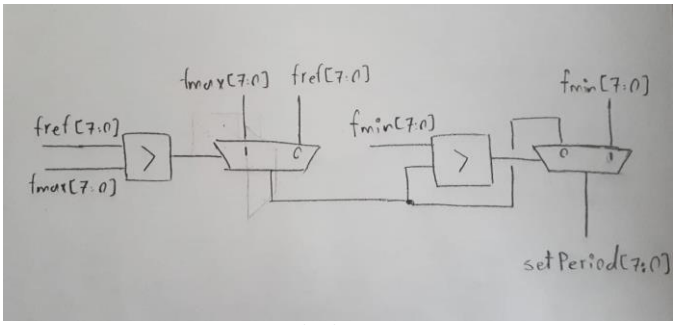

Fig. 2 RTL design for fspec-range-spec unit

III. FREQUENCY REGULATOR UNIT

This section is the main processing unit. It inputs the T flip flop output (PSI) from the clock divider unit (implemented in experiment #1). It counts PSI's high pulse duration (in terms of number clocks). If this counted duration is larger than setPeriod[7:0], it increments the adjustedLoad[7:0] for counters in clock divider by 1 unit so they count less number of units so the duration of high pulse on PSI would become closer to setPeriod[7:0]. If this counted duration is less than setPeriod[7:0], it decrements the adjustedLoad[7:0] for counters in clock divider by 1 unit so they count more number of units so the high pulse duration of PSI would become closer to setPeriod[7:0]. setPeriod[7:0] is the desired high pulse duration of PSI in terms of number of clocks.

This unit's clock is the FPGA's clock which we set to be 50MHz. It changes the value (increment or decrement) of adjustedDiv[7:0] when seeing a falling edge on PSI. It starts counting the duration when seeing a rising edge on PSI.

```
always @(posedge clk, posedge rst)
  if (rst)
    oldpsi <= 1'b0;
  else
    oldpsi <= psi;

always @(posedge clk or posedge rst) begin
      if (rst == 1'b1)
          count <= 16'b0000000000000000;
      else begin
          case ({oldpsi, psi})
              2'b00: count <= count;
              2'b01: count <= 16'b0000000000000000;
              2'b11: count <= count + 16'b0000000000000001;
              2'b10: count <= count;
          endcase
      end
  end


  always @(oldpsi or psi or count or setPeriod) begin
      {inc, dec} = 2'b00;
      if ({oldpsi, psi} == 2'b10) begin
          if (count + 16'b0000000000000001 > {8'b00000000, setPeriod})
              inc = 1'b1;
          if (count + 16'b0000000000000001 < {8'b00000000, setPeriod})
              dec = 1'b1;
      end
  end

  always @(posedge clk or posedge rst) begin
      if (rst == 1'b1)
          adjustedDiv <= 8'b0111_1111;
      else if ({oldpsi, psi} == 2'b10) begin
          if (inc == 1'b1)
              adjustedDiv <= adjustedDiv + 8'b00000001;
          if (dec == 1'b1)
              adjustedDiv <= adjustedDiv - 8'b00000001;
      end
  end
```

Fig. 3 Verilog description of Frequency Regulator unit

IV. CLOCK DIVIDER UNIT

This unit is implemented using two 74193 IC's. They are cascaded and used to build a counter with parallel load inputs coming from the frequency regulator unit. The clock of this unit is 20MHz generated by a ring oscillator. Whenever the counter's reaches 255 (11111111), the carry out of the MSB counter becomes 1 and there is a pulse on the clock of the T Flip Flop, therefore it toggles. Afterwards, the counters start counting from the loaded values.

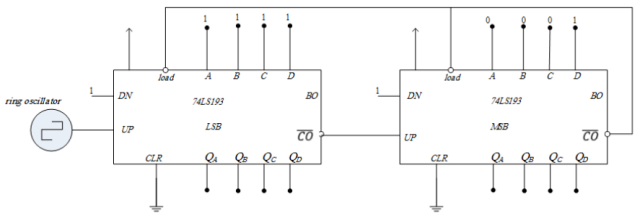

Fig 4. Frequency divider using 74193

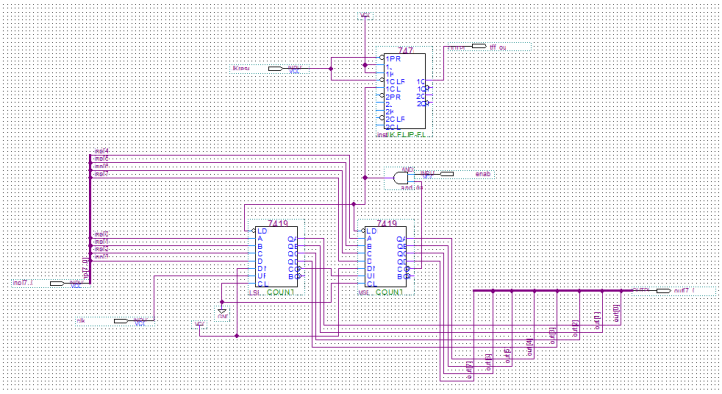

Fig 5. Schematic of Clock Divider unit in Quartus

V. MODELSIM SIMULATION OF THE ADJUSTER

Two scenarios are used for testing the adjuster circuit after synthesis. In both scenarios the frequency of the clock divider's unit clock (ring oscillator) is 20MHz and the frequency of the frequency regulator unit (main processing unit) is 48.78MHz (T = 20.5 ns).

In the waveforms, the clock generated by the ring oscillator for the clock divider unit is called is named clk_divider and the clock used for the frequency regulator unit is named clk_regulator. The counted value by the counters in the clock divider section (Experiment #1) is called cnt.

In the first scenario setPeriod[7:0] is set to 125. The adjusted load value finally reaches 203. The minor difference with 205 is caused because the clock is 48.78MHz instead of 50MHz.

$T_{RING\ OSCILLATOR} = 1 / f_{RING\ OSCILLATOR} = 1 / 20 = 50ns$
$T_{FPGA} = 1 / f_{FPGA} = 1 / 48.78 = 20.5ns$

Desired Frequency = 1 / (125 × 20.5) = 2562.5 KHz
Desired Period = 125 × 20.5 = 2562.5ns

If we denote the number of units the counter has to count by N then we have:

$$T_{RING\ OSCILLATOR} \times N = 2562.5$$
$$N = 51.25$$

The counter needs to count N = 52 units, therefore parallel load will be adjusted to 255 – 52 = 203 after given enough time.

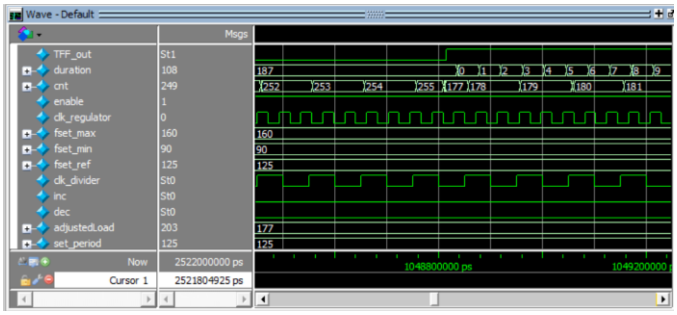

Fig 6. Whenever there is a rising edge on PSI (TFF output), the duration counter starts counting from *adjustedLoad*.

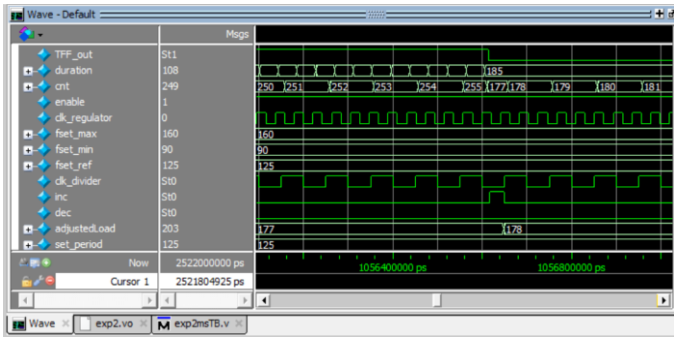

Fig 7. Whenever there is a falling edge on PSI (TFF Out) the value of duration is compared with *setPeriod*. In this figure 185 > 125 therefore it needs to increment *adjustedLoad*. As you can see an *inc* signal is issued to 1 in this case.

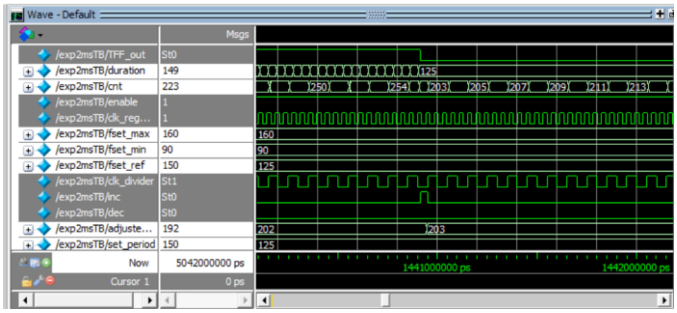

Fig 8. The process in figure 6 is continued until finally the counted duration becomes *setPeriod*. This happens when parallel load is 203.

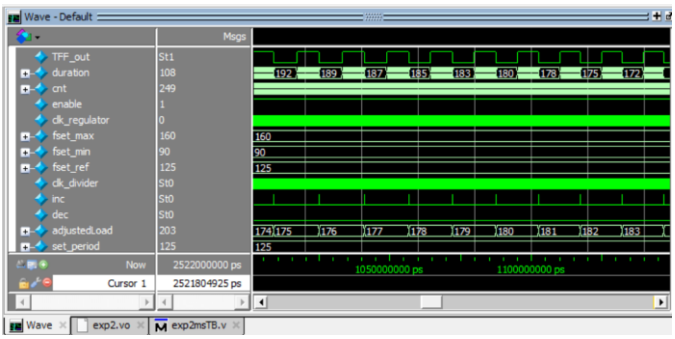

Fig 9. An overall view of the process. *Duration* is decreasing until it reaches *setPeriod*.

In the second scenario, the pre-processing unit is verified. A value outside the valid range (170) is given. As it can be seen in the waveforms in Figure 10, *setPeriod* is set to be *fset-max* which is 160.

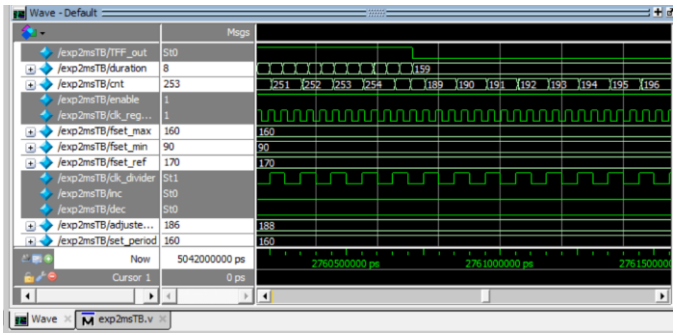

Fig 10. Freq-Range-Spec operating properly

REFERENCES

[1] SN7407 Datasheet, Provided by Texas Instruments
[2] DM74LS191 Datasheet, Provided by Fairchild Semiconductor