CS M152A - Introduction to Digital Design Lab

Project 2: Clock Design Methodology

Arnav Garg - 304911796

Introduction and Background:

This lab focuses on designing clocks using modern counters and variations of modern counters. Clocks play an important role in digital systems because they make it easier to test very specific aspects of circuits, especially in cases where they need to observe waveforms at unconventional instances. While this could theoretically be done using just the system clock, the work becomes a lot harder. Creating clocks lets us streamline this entire aspect, allowing simulations to run at a variety of different clock speeds. Clocking is also particularly popular in synchronous data transmission, that is, it tends to be used a lot in communicating signals between devices, such as in the I2C bus. These components typically make up time-based devices like traffic lights.

In particular, this lab makes use of the system clock and requires us to generate four different variations of clocks: a divide by 2^n clock, an even division divide-by-28 clock, and odd-division divide-by-5 clock, and a glitchy counter that is intended to help us explore pulse/strobe/flag. In the process of creating these 4 variations of clocks, we end up performing 10 different tasks, each of which focuses on a specific aspect of designing a clock, and many of which help incrementally build the 4 required clocks.

In general, clocks are a sequential circuit since the output from the previous state is required to move to the next state. This intuitively explains why counters are well suited for this task - we can keep incrementing a counter, which essentially takes the previous state and adds one to it. If we coordinate the increments of the counter with the positive or negative edge of the system clock, we can easily simulate a clock using the counter.

Design Description + Simulation:

At a very high level, all required modules take a system clock and reset signal as inputs, and output the derived clock. There is one top-level module, *clock_gen*, that creates instances of 4 different modules: *clock_div_two*, *clock_div_twenty_eight*, *clock_div_five*, and *clock_strobe*. The description for these 4 submodules is provided in the table below.

clock_gen		
Divide by 2^n Clock	clock_div_two	The submodule exploring clock division by power of 2
Even Division Clock	clock_div_twenty_eight	The submodule exploring even clock division
Odd Division Clock	clock_div_five	The submodule exploring odd clock division
Glitchy Counter	clock_strobe	The submodule exploring pulse/strobe/flag

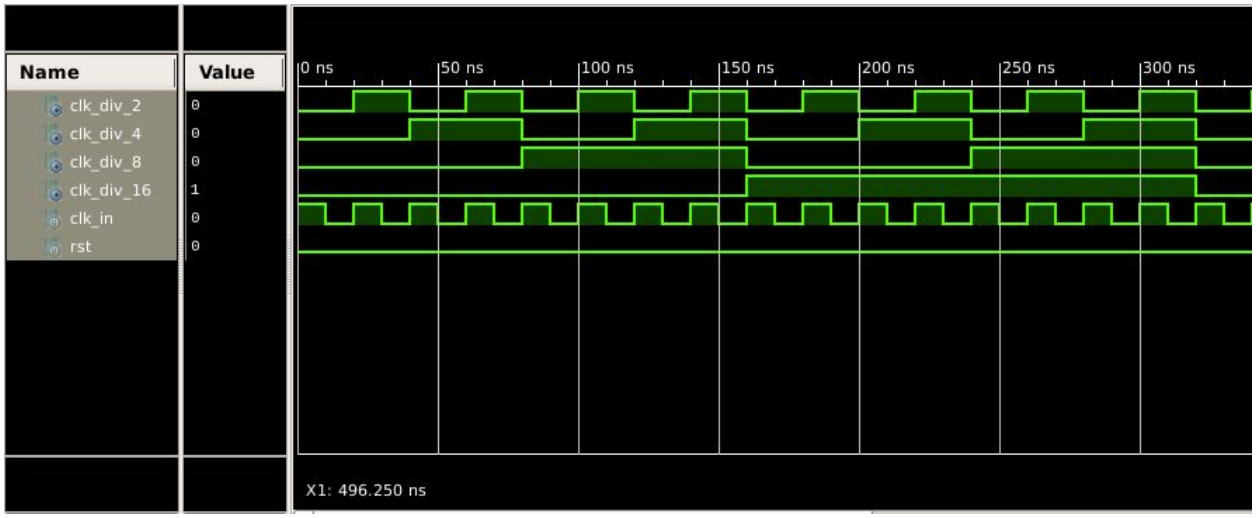
1. Clock Divider by Power of 2s:

This submodule uses a 4-bit counter to generate 4 different clocks: clock_div_2, clock_div_4, clock_div_8, and clock_div_16. The table below contains the set of inputs and outputs for this module.

Inputs	Type
clk_in	1-bit wire
rst	1-bit wire
clk_div_2	1-bit reg
clk_div_4	1-bit reg
clk_div_8	1-bit reg
clk_div_16	1-bit reg

The basic idea for this module is that if we take a 4-bit counter, it has a range from 0 to 15. Therefore, if we observe the nature of the 4-bit counter, the Lease Significant bit (LSB) flips every second clock cycle, the 2nd LSB flips once every four clock cycles, the 3rd LSB flips every 8 clock cycles, and the MSB flips every 16 clock cycles. Therefore, if we assign clk_div_2 to the LSB, clk_div_4 to the second LSB, clk_div_8 to the third LSB and clk_div_16 to the MSB, we will see that they mimic a clock divide by powers of 2. That is, clk_div_2 will flip every 2 cycles of the system clock. This is reflected in the simulation output for Task 1.

Task 1:



The simulation clearly illustrates that clk_div_2 flip at the end of each system clock cycle, and clk_div_16 flips at the end of every 8 clock cycles. Therefore, one entire period of the clk_div_16 takes 16 system clock cycles, which is also seen in the simulation.

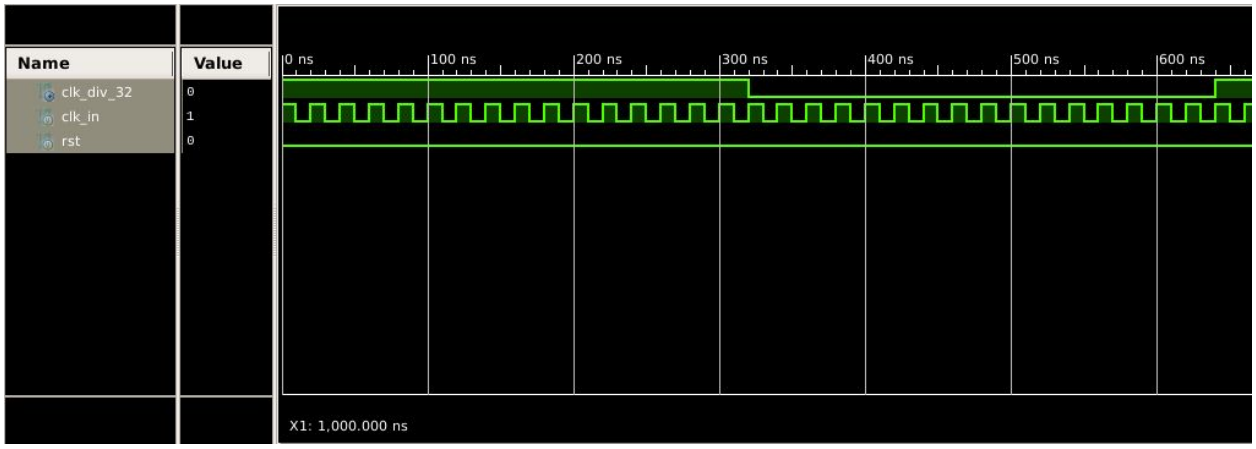
2. Even Division Clock Using Counters:

For this section, we had to use a 4-bit counter to simulate both a divide by 32 and a divide by 28 clocks. This section works very similarly to submodule 1, where it makes use of a 4-bit counter to simulate even division. It also works on the same principle, that is, we can flip the bit when the 4-bit counter equals a particular value. The bit is flipped at half the duration, that is, if we want a 32-bit divider, we want to flip the output clock when the counter resets, and if we want a 28-bit divider, then we must flip the bit when the counter is 14.

Below is a table showing the outputs and inputs into the submodule.

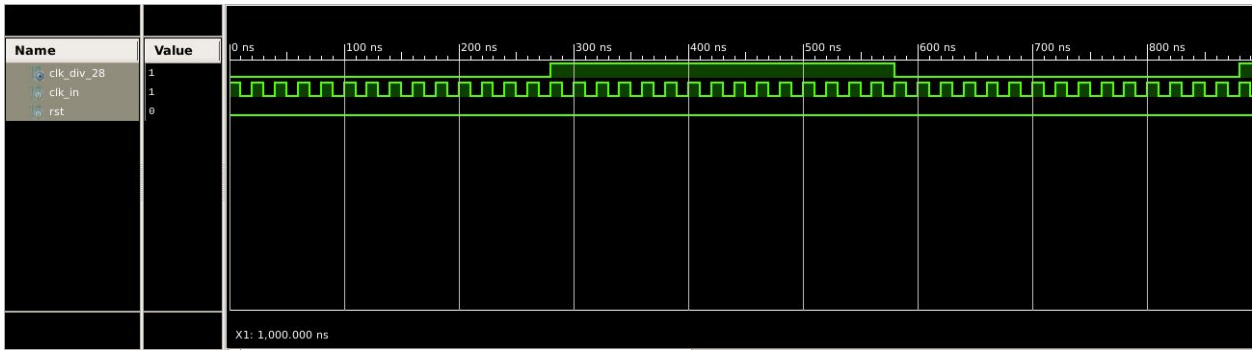
Inputs	Type
clk_in	1-bit wire
rst	1-bit wire
clk_div_28/clk_div_32	1-bit reg

Task 2:



For task 2, we had the same system clock, *clk_in*, and *rst* as inputs, and had *clk_div_32* as output. The 4-bit counter used internally in the module counts from 0 to 15 and flips *clk_div_32* every time the counter resets to zero when the 4-bit register overflows. This results in the output clock flipping every 16 counts of the system clock, and the output clock completes one clock cycle every 32 cycles of the input/system clock.

Task 3:



For task 3, we had the same system clock, *clk_in*, and *rst* as inputs, and had *clk_div_28* as output. The 4-bit counter used internally in the module counts from 0 to 13 and flips *clk_div_28* every time the counter is equal to 14 (4'b1110) when the 4-bit register overflows. This results in the output clock flipping every 14 counts of the system clock, and the output clock completes one clock cycle every 28 cycles of the input/system clock.

3. Odd Division Clock Using Counters:

For submodule three, we were required to create a 33% duty cycle clock using if statements and counters. In general, the relationship between duty cycle and time period is:

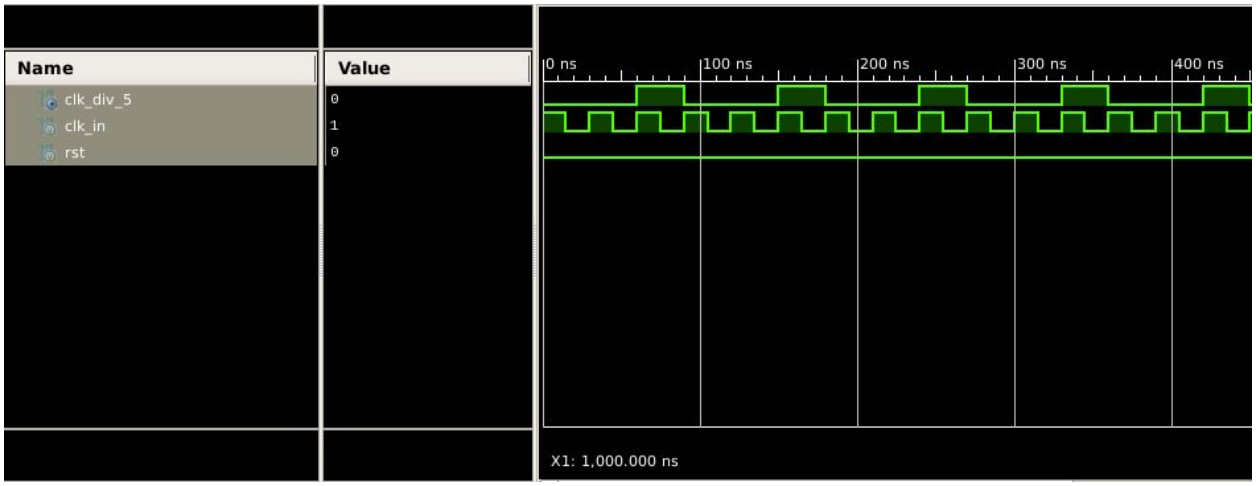
$$Duty\ cycle\ = time\ period\ on\ /\ total\ time\ period$$

Therefore, to achieve a duty cycle of ⅓ or 33%, we need the clock to turn on for ⅓rd of the total duration in one clock period. Specifically, we are required to use a divide by 3 clock and simulate an "odd" division.

Below is a table showing the outputs and inputs into the submodule.

Inputs	Type
clk_in	1-bit wire
rst	1-bit wire
clk_div_5	1-bit reg (2-bit for Task 5)

Task 4:

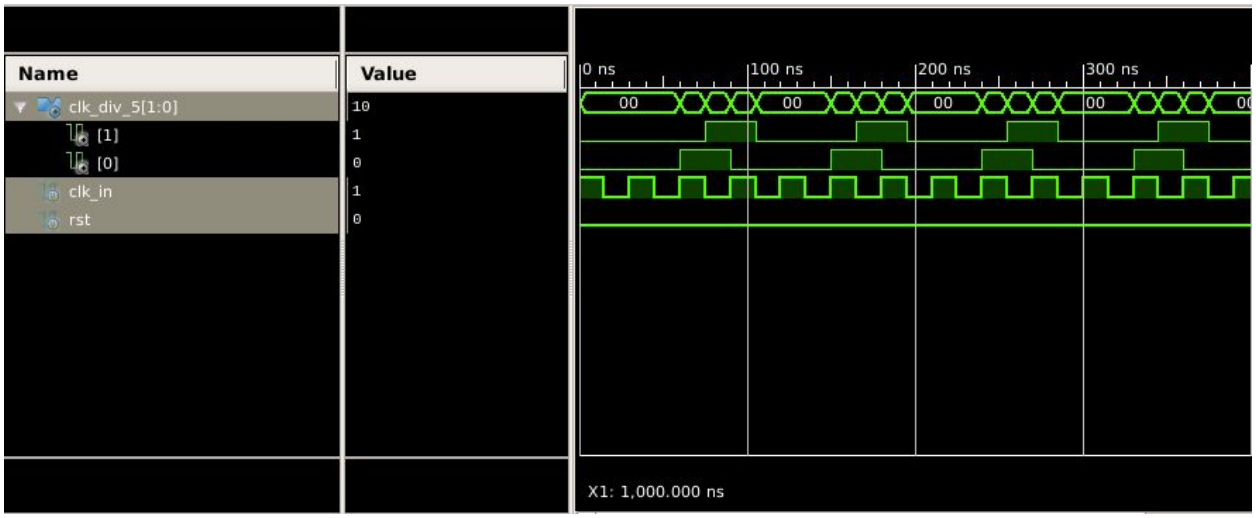


In this simulation, we see that clk_div_5 (which is really clk_div_3) goes high for 1/3rd of the total duration of the time and stays low/off for 2/3rd of the duration. The way this was done is by using the following code:

```
reg [1:0] a = 2'b00;


always@(posedge clk_in, posedge rst)
begin
    if (rst)
        a <= 2'b00;
    else if (a == 2'b10)
    begin
        a <= 2'b00;
        clk_div_5 <= ~clk_div_5; //functions like a t flip-flop
    end
    else
        a <= a + 1'b1;
end
```

Here, we essentially use a 2-bit counter to count from 0 to 2. This results in a total count of 3. To make sure the output clock is only turned on/high for ⅓ counts, we flip the bit every time the counter strikes 2. when it goes back to zero, the output bit is flipped back to zero. Therefore, this sumlates a divide by 3 clock with a 33% duty cycle.

Task 5:



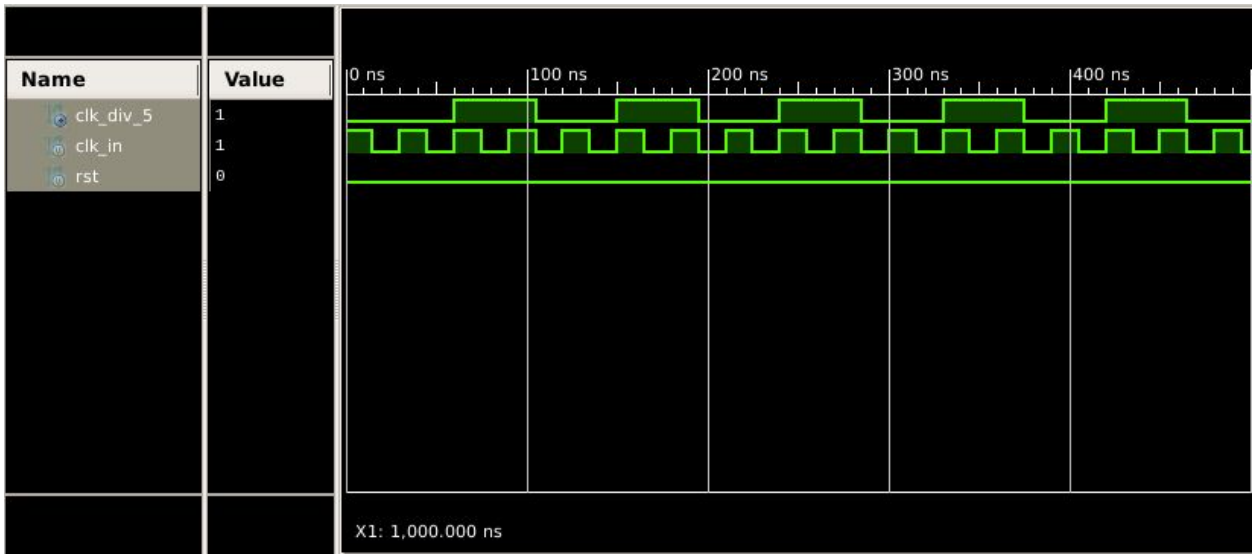
For task 5, we repeat the code for Task 4. However, we ensure that we create a separate a counter that would be triggered with the falling edge of the system clock, and change the output, clk_div_4 (actually clk_div_3) to a 2-bit bus, where the first bit represents the rising edge, while the MSB represents the falling edge. The code segment used was as follows:

```
input clk_in, rst;
output reg [1:0] clk_div_5 = {1'b0, 1'b0};

reg [1:0] a = 2'b00;
reg [1:0] b = 2'b00;

always@(posedge clk_in, posedge rst)
begin
    if (rst)
        a <= 2'b00;
    else if (a == 2'b10)
    begin
        a <= 2'b00;
        clk_div_5[0] <= ~clk_div_5[0];
    end
    else
    begin
        a <= a + 1'b1;
        clk_div_5[0] <= 1'b0;
    end
end
```

```
always@(negedge clk_in, posedge rst)
begin
    if (rst)
        b <= 2'b00;
    else if (b == 2'b10)
    begin
        b <= 2'b00;
        clk_div_5[1] <= ~clk_div_5[1];
    end
    else
    begin
        b <= b + 1'b1;
        clk_div_5[1] <= 1'b0;
    end
end
```

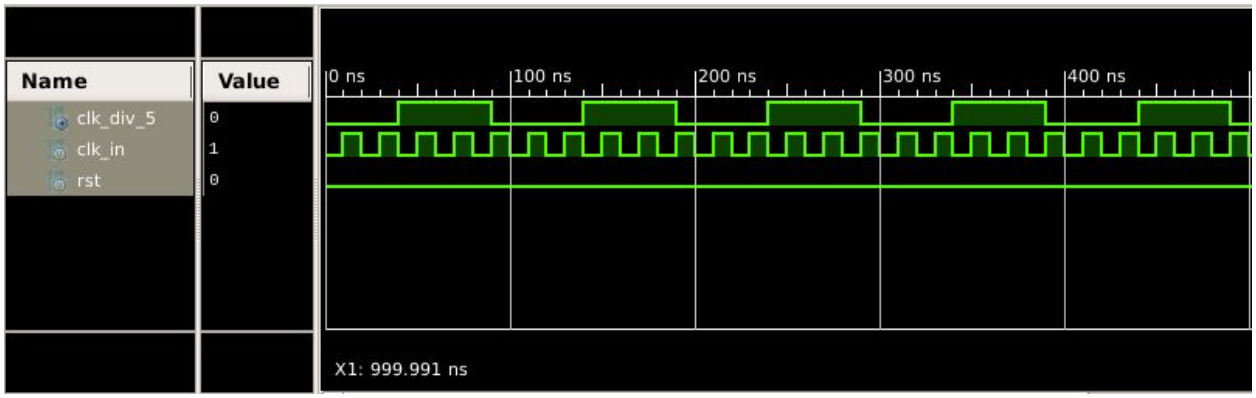
Therefore, this simulates both the positive nad negative triggred divide by 3 coutner working at the same time, side by side. The waveforms indicate that both of them are triggered separately based on whether they are triggered by the rising or falling edge of the sytem clock.

Task 6:



For task 6, we had to take the logical OR of the outputs of the clocks triggered by the falling and rising edge of the system clock respectively. Because of how they work, there is interference and this results in a duty cycle of 50% using two simultaneous divide by 3 clocks triggered on different edges. Half the clock cycle, in this case, is a system clock period of 1.5 cycles, thereby a full period is 3 system clock cycles, resulting in a divide by 3 counter.

Task 7:



For Task 7, we took the idea from Task 6 and replaced the divide by 3 counter to a divide by 5 counter. This made it a bit trickier since it no longer could take the logical OT of the outputs of the two individual divide by 3 counters triggered by the falling and rising edge of the system clock respectively. Instead, we need to take the logical OR of whether each of the two outputs was greater than 2'b10, or 2. This is a result of taking 5 and performing a right shift by 1, so 5 >> 1 becomes 101 >> 2 which is 010 which is essentially an (N-1)/2 division. The reason we had to do this is to maintain a 50% duty cycle, otherwise, this would result in a 20% duty cycle, which was incorrect.

4. Pulse/Strobes:

Below is a table showing the outputs and inputs into the submodule.

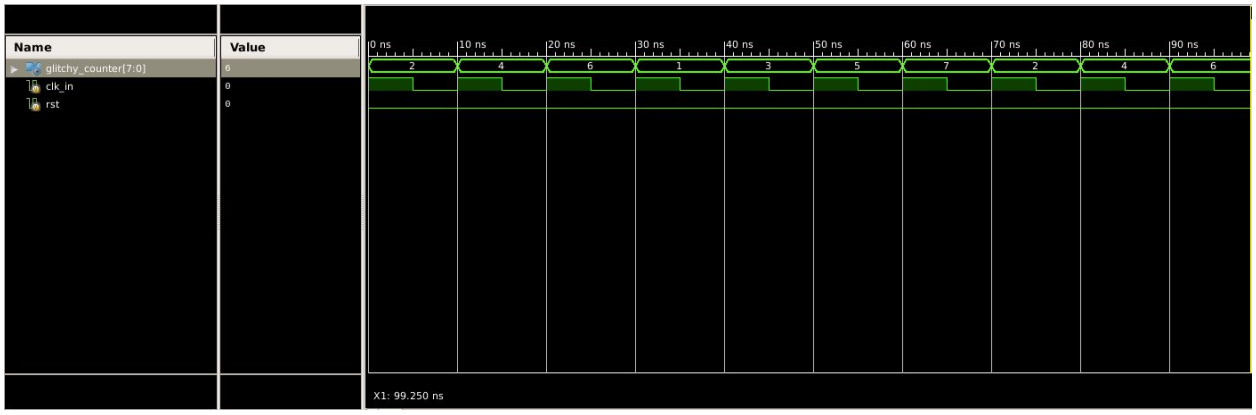
Inputs	Type
clk_in	1-bit wire
rst	1-bit wire
glitchy_counter	8-bit reg

Task 8:

For task 8, we create a divide by 100 counter in a very similar manner to the divide by 3 counter in Task 4. The difference here is that we want a duty cycle of 1%. The way to produce is to flip the bit only once when the counter reaches 100, and then to reset the counter back to 0. For this, we can use an 8-bit register, which has a maximum value of 128. To simulate a 100MHz clock, the test bench was set to a timescale of 1ns/ps and the rest duration was 10ns, which simulates a 100MHz clock. To ensure that we could switch the output clock every time the

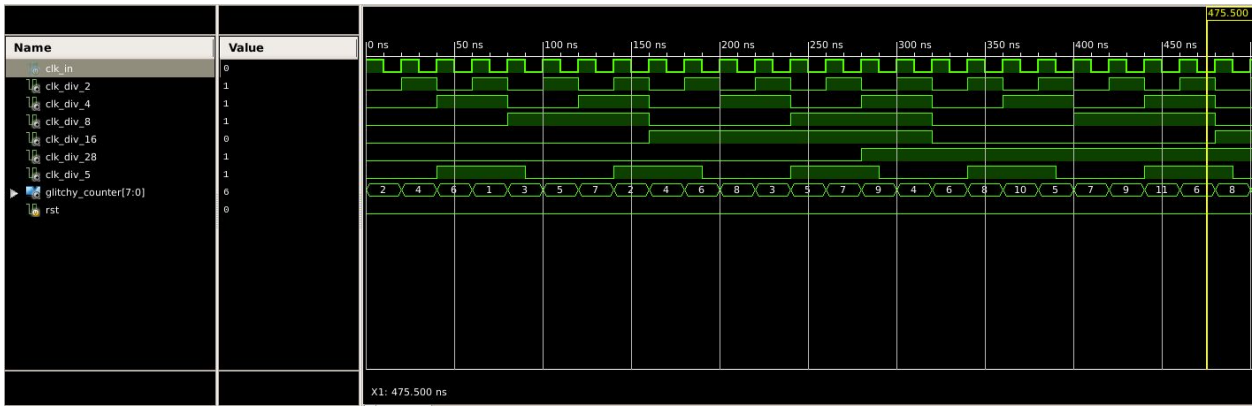
divide-by-100 pulse is active, I used an if statement that flipped the current value of the glictch_counter in the output. This resulted in an output clock with a 50% duty cycle that is running at 500Khz and is divide-by-200. To confirm this, we can see that the period of the output clock is 2us, which corresponds to a frequency of 500KHz.

Task 9:



For this part of the project, we were required to use a divide-by-4 clock to generate an 8-bit counter that counts up by 2 on every positive edge of the master clock but subtracts by 5 on every strobe. For this, we make use of the divide_by_two module but specifically make use of the divide_by_4 output from the register. We want to increment the output by 2 every time divide_by_4 is 0, but we subtract 5 every time divide_by_4 is 1. The result is that, after 4 clock cycles of the master clock, the output is decremented by 5. However, because we increment by 6 and then subtract by 5, the result is that every 4th cycle results in the previous value 4 cycles ago + 1. It's also important to note that we must keep track of the previous flip separately, otherwise it will result in more subtractions than we like.

Task 10:



Task 10 provides a high-level overview of Tasks 1, 4, 7, and 9. We can see that all waveforms work as expected.

Conclusion:

This lab was extremely interesting and it helped me better understand how clocks are generated, especially with different duty cycles using a variety of different methods including divide by odd and even numbers. I think the control different granularity of clocks gives allows for very specific forms of testing that will help FPGA developers make and simulate more edge cases in their designs, and I can see the merit of making all of these clocks.

I particularly found Task 8 challenging since it was difficult to confirm the validity of the waveform. I believe that this project had a lot of different tasks that required a lot of effort. It would have been easier to create a different module for each task to keep things more organized, but I completely understand why we just needed to create 4 different modules.

Synthesis Report:

```
Release 14.7 - xst P.20131013 (lin64)
Copyright (c) 1995-2013 Xilinx, Inc.  All rights reserved.
-->
Parameter TMPDIR set to xst/projnav.tmp


Total REAL time to Xst completion: 0.00 secs
Total CPU time to Xst completion: 0.04 secs

-->
Parameter xsthdpdir set to xst


Total REAL time to Xst completion: 0.00 secs
Total CPU time to Xst completion: 0.04 secs

-->
Reading design: clock_gen.prj

TABLE OF CONTENTS
  1) Synthesis Options Summary
  2) HDL Parsing
  3) HDL Elaboration
  4) HDL Synthesis
       4.1) HDL Synthesis Report
  5) Advanced HDL Synthesis
       5.1) Advanced HDL Synthesis Report
  6) Low Level Synthesis
  7) Partition Report
  8) Design Summary
       8.1) Primitive and Black Box Usage
       8.2) Device utilization summary
       8.3) Partition Resource Summary
       8.4) Timing Report
```

```
=========================================================================
*                      Synthesis Options Summary                        *
=========================================================================
---- Source Parameters
Input File Name                    : "clock_gen.prj"
Ignore Synthesis Constraint File   : NO

---- Target Parameters
Output File Name                   : "clock_gen"
Output Format                      : NGC
Target Device                      : xc7a100t-3-csg324

---- Source Options
Top Module Name                    : clock_gen
Automatic FSM Extraction           : YES
FSM Encoding Algorithm             : Auto
Safe Implementation                : No
FSM Style                          : LUT
RAM Extraction                     : Yes
RAM Style                          : Auto
ROM Extraction                     : Yes
Shift Register Extraction          : YES
ROM Style                          : Auto
Resource Sharing                   : YES
Asynchronous To Synchronous        : NO
Shift Register Minimum Size        : 2
Use DSP Block                      : Auto
Automatic Register Balancing       : No

---- Target Options
LUT Combining                      : Auto
Reduce Control Sets                : Auto
Add IO Buffers                     : YES
Global Maximum Fanout              : 100000
Add Generic Clock Buffer(BUFG)     : 32
Register Duplication               : YES
Optimize Instantiated Primitives   : NO
Use Clock Enable                   : Auto
Use Synchronous Set                : Auto
Use Synchronous Reset              : Auto
Pack IO Registers into IOBs        : Auto
Equivalent register Removal        : YES

---- General Options
Optimization Goal                  : Speed
Optimization Effort                : 1
Power Reduction                    : NO
```

```
Keep Hierarchy                     : No
Netlist Hierarchy                  : As_Optimized
RTL Output                         : Yes
Global Optimization                : AllClockNets
Read Cores                         : YES
Write Timing Constraints           : NO
Cross Clock Analysis               : NO
Hierarchy Separator                : /
Bus Delimiter                      : <>
Case Specifier                     : Maintain
Slice Utilization Ratio            : 100
BRAM Utilization Ratio             : 100
DSP48 Utilization Ratio            : 100
Auto BRAM Packing                  : NO
Slice Utilization Ratio Delta      : 5

=========================================================================


=========================================================================
*                            HDL Parsing                                *
=========================================================================
Analyzing Verilog file "/home/ise/Project3/clock_div_two.v" into library work
Parsing module <clock_div_two>.
Analyzing Verilog file "/home/ise/Project3/clock_strobe.v" into library work
Parsing module <clock_strobe>.
Analyzing Verilog file "/home/ise/Project3/clock_div_twenty_eight.v" into
library work
Parsing module <clock_div_twenty_eight>.
Analyzing Verilog file "/home/ise/Project3/clock_div_five.v" into library work
Parsing module <clock_div_five>.
Analyzing Verilog file "/home/ise/Project3/clock_gen.v" into library work
Parsing module <clock_gen>.

=========================================================================
*                            HDL Elaboration                            *
=========================================================================

Elaborating module <clock_gen>.

Elaborating module <clock_div_two>.

Elaborating module <clock_div_twenty_eight>.

Elaborating module <clock_div_five>.

Elaborating module <clock_strobe>.
WARNING:HDLCompiler:1127 - "/home/ise/Project3/clock_strobe.v" Line 38:
Assignment to clk_div_2 ignored, since the identifier is never used
WARNING:HDLCompiler:1127 - "/home/ise/Project3/clock_strobe.v" Line 40:
Assignment to clk_div_8 ignored, since the identifier is never used
WARNING:HDLCompiler:1127 - "/home/ise/Project3/clock_strobe.v" Line 41:
Assignment to clk_div_16 ignored, since the identifier is never used
```

```
WARNING:HDLCompiler:413 - "/home/ise/Project3/clock_strobe.v" Line 51: Result
of 32-bit expression is truncated to fit in 8-bit target.
WARNING:HDLCompiler:413 - "/home/ise/Project3/clock_strobe.v" Line 53: Result
of 9-bit expression is truncated to fit in 8-bit target.

=========================================================================
*                            HDL Synthesis                              *
=========================================================================

Synthesizing Unit <clock_gen>.
    Related source file is "/home/ise/Project3/clock_gen.v".
    Summary:
      no macro.
Unit <clock_gen> synthesized.

Synthesizing Unit <clock_div_two>.
    Related source file is "/home/ise/Project3/clock_div_two.v".
    Found 4-bit register for signal <a>.
    Found 1-bit register for signal <clk_div_2>.
    Found 1-bit register for signal <clk_div_4>.
    Found 1-bit register for signal <clk_div_8>.
    Found 1-bit register for signal <clk_div_16>.
    Found 4-bit adder for signal <a[3]_GND_2_o_add_4_OUT> created at line 65.
    Summary:
      inferred   1 Adder/Subtractor(s).
      inferred   8 D-type flip-flop(s).
Unit <clock_div_two> synthesized.

Synthesizing Unit <clock_div_twenty_eight>.
    Related source file is "/home/ise/Project3/clock_div_twenty_eight.v".
    Found 4-bit register for signal <a>.
    Found 1-bit register for signal <clk_div_28>.
    Found 4-bit adder for signal <a[3]_GND_3_o_add_0_OUT> created at line 43.
    Summary:
      inferred   1 Adder/Subtractor(s).
      inferred   5 D-type flip-flop(s).
      inferred   1 Multiplexer(s).
Unit <clock_div_twenty_eight> synthesized.

Synthesizing Unit <clock_div_five>.
    Related source file is "/home/ise/Project3/clock_div_five.v".
    Found 3-bit register for signal <clk_neg>.
    Found 3-bit register for signal <clk_pos>.
    Found 3-bit adder for signal <clk_pos[2]_GND_4_o_add_1_OUT> created at line
40.
    Found 3-bit adder for signal <clk_neg[2]_GND_4_o_add_5_OUT> created at line
50.
    Found 3-bit comparator greater for signal <GND_4_o_clk_pos[2]_LessThan_9_o>
created at line 53
    Found 3-bit comparator greater for signal
<GND_4_o_clk_neg[2]_LessThan_10_o> created at line 53
    Summary:
      inferred   2 Adder/Subtractor(s).
      inferred   6 D-type flip-flop(s).
```

```
      inferred   2 Comparator(s).
      inferred   2 Multiplexer(s).
Unit <clock_div_five> synthesized.

Synthesizing Unit <clock_strobe>.
    Related source file is "/home/ise/Project3/clock_strobe.v".
INFO:Xst:3210 - "/home/ise/Project3/clock_strobe.v" line 35: Output port
<clk_div_2> of the instance <clk2> is unconnected or connected to loadless
signal.
INFO:Xst:3210 - "/home/ise/Project3/clock_strobe.v" line 35: Output port
<clk_div_8> of the instance <clk2> is unconnected or connected to loadless
signal.
INFO:Xst:3210 - "/home/ise/Project3/clock_strobe.v" line 35: Output port
<clk_div_16> of the instance <clk2> is unconnected or connected to loadless
signal.
    Found 1-bit register for signal <clk_prev>.
    Found 8-bit register for signal <glitchy_counter>.
    Found 8-bit adder for signal <glitchy_counter[7]_GND_5_o_add_3_OUT> created
at line 53.
    Found 8-bit subtractor for signal <GND_5_o_GND_5_o_sub_3_OUT<7:0>> created
at line 51.
    Found 1-bit comparator not equal for signal <n0000> created at line 50
    Summary:
      inferred   1 Adder/Subtractor(s).
      inferred   9 D-type flip-flop(s).
      inferred   1 Comparator(s).
      inferred   1 Multiplexer(s).
Unit <clock_strobe> synthesized.

=========================================================================
HDL Synthesis Report

Macro Statistics
# Adders/Subtractors                                   : 6
 3-bit adder                                           : 2
 4-bit adder                                           : 3
 8-bit addsub                                          : 1
# Registers                                            : 10
 1-bit register                                        : 2
 3-bit register                                        : 2
 4-bit register                                        : 5
 8-bit register                                        : 1
# Comparators                                          : 3
 1-bit comparator not equal                            : 1
 3-bit comparator greater                              : 2
# Multiplexers                                         : 4
 3-bit 2-to-1 multiplexer                              : 2
 4-bit 2-to-1 multiplexer                              : 1
 8-bit 2-to-1 multiplexer                              : 1

=========================================================================
INFO:Xst:1767 - HDL ADVISOR - Resource sharing has identified that some
arithmetic operations in this design can share the same physical resources for
```

```
reduced device utilization. For improved clock frequency, you may try to
disable resource sharing.

=========================================================================
*                        Advanced HDL Synthesis                         *
=========================================================================

WARNING:Xst:2677 - Node <clk_div_16> of sequential type is unconnected in block
<clk2>.
WARNING:Xst:2677 - Node <clk_div_4> of sequential type is unconnected in block
<clk2>.
WARNING:Xst:2677 - Node <clk_div_2> of sequential type is unconnected in block
<clk2>.

Synthesizing (advanced) Unit <clock_div_five>.
The following registers are absorbed into counter <clk_neg>: 1 register on
signal <clk_neg>.
The following registers are absorbed into counter <clk_pos>: 1 register on
signal <clk_pos>.
Unit <clock_div_five> synthesized (advanced).

Synthesizing (advanced) Unit <clock_div_twenty_eight>.
The following registers are absorbed into counter <a>: 1 register on signal
<a>.
Unit <clock_div_twenty_eight> synthesized (advanced).

Synthesizing (advanced) Unit <clock_div_two>.
The following registers are absorbed into counter <a>: 1 register on signal
<a>.
Unit <clock_div_two> synthesized (advanced).

Synthesizing (advanced) Unit <clock_strobe>.
The following registers are absorbed into accumulator <glitchy_counter>: 1
register on signal <glitchy_counter>.
Unit <clock_strobe> synthesized (advanced).

=========================================================================
Advanced HDL Synthesis Report

Macro Statistics
# Counters                                             : 5
 3-bit up counter                                     : 2
 4-bit up counter                                     : 3
# Accumulators                                         : 1
 8-bit updown accumulator                             : 1
# Registers                                            : 10
 Flip-Flops                                           : 10
# Comparators                                          : 3
 1-bit comparator not equal                           : 1
 3-bit comparator greater                             : 2

=========================================================================

=========================================================================
```

```
*                          Low Level Synthesis                          *
================================================================================
WARNING:Xst:2677 - Node <task_four/clk2/clk_div_16> of sequential type is
unconnected in block <clock_gen>.
WARNING:Xst:2677 - Node <task_four/clk2/clk_div_4> of sequential type is
unconnected in block <clock_gen>.
WARNING:Xst:2677 - Node <task_four/clk2/clk_div_2> of sequential type is
unconnected in block <clock_gen>.
INFO:Xst:2146 - In block <clock_gen>, Counter <task_one/a> <task_four/clk2/a>
are equivalent, XST will keep only <task_one/a>.
INFO:Xst:2261 - The FF/Latch <task_one/clk_div_8> in Unit <clock_gen> is
equivalent to the following FF/Latch, which will be removed :
<task_four/clk2/clk_div_8>

Optimizing unit <clock_gen> ...

Mapping all equations...
Building and optimizing final netlist ...
Found area constraint ratio of 100 (+ 5) on block clock_gen, actual ratio is 0.

Final Macro Processing ...

================================================================================
Final Register Report

Macro Statistics
# Registers                                            : 28
 Flip-Flops                                            : 28

================================================================================

================================================================================
*                            Partition Report                                  *
================================================================================

Partition Implementation Status
-------------------------------

  No Partitions were found in this design.

-------------------------------

================================================================================
*                             Design Summary                                   *
================================================================================

Top Level Output File Name         : clock_gen.ngc

Primitive and Black Box Usage:
------------------------------
# BELS                             : 30
#      INV                         : 3
#      LUT2                        : 4
#      LUT3                        : 7
```

```
#      LUT4                          : 7
#      LUT5                          : 2
#      LUT6                          : 7
# FlipFlops/Latches                  : 28
#      FD                            : 2
#      FDC                           : 14
#      FDE                           : 4
#      FDR                           : 8
# Clock Buffers                      : 1
#      BUFGP                         : 1
# IO Buffers                         : 15
#      IBUF                          : 1
#      OBUF                          : 14

Device utilization summary:
---------------------------

Selected Device : 7a100tcsg324-3


Slice Logic Utilization:
 Number of Slice Registers:              28  out of  126800     0%
 Number of Slice LUTs:                   30  out of  63400     0%
    Number used as Logic:                30  out of  63400     0%

Slice Logic Distribution:
 Number of LUT Flip Flop pairs used:     34
   Number with an unused Flip Flop:       6  out of     34    17%
   Number with an unused LUT:             4  out of     34    11%
   Number of fully used LUT-FF pairs:    24  out of     34    70%
   Number of unique control sets:         5

IO Utilization:
 Number of IOs:                          16
 Number of bonded IOBs:                  16  out of    210     7%

Specific Feature Utilization:
 Number of BUFG/BUFGCTRLs:                1  out of     32     3%

---------------------------
Partition Resource Summary:
---------------------------

  No Partitions were found in this design.

---------------------------


=========================================================================
Timing Report

NOTE: THESE TIMING NUMBERS ARE ONLY A SYNTHESIS ESTIMATE.
      FOR ACCURATE TIMING INFORMATION PLEASE REFER TO THE TRACE REPORT
      GENERATED AFTER PLACE-and-ROUTE.
```

```
Clock Information:
------------------
+-----------------------------------+------------------------+-------+
Clock Signal                        | Clock buffer(FF name)  | Load  |
+-----------------------------------+------------------------+-------+
clk_in                              | BUFGP                  | 28    |
+-----------------------------------+------------------------+-------+

Asynchronous Control Signals Information:
----------------------------------------
No asynchronous control signals found in this design

Timing Summary:
---------------
Speed Grade: -3

   Minimum period: 1.570ns (Maximum Frequency: 637.024MHz)
   Minimum input arrival time before clock: 0.884ns
   Maximum output required time after clock: 1.445ns
   Maximum combinational path delay: No path found

Timing Details:
---------------
All values displayed in nanoseconds (ns)

=========================================================================
Timing constraint: Default period analysis for Clock 'clk_in'
  Clock period: 1.570ns (frequency: 637.024MHz)
  Total number of paths / destination ports: 114 / 28
-------------------------------------------------------------------------
Delay:               1.570ns (Levels of Logic = 2)
  Source:            task_four/glitchy_counter_3 (FF)
  Destination:       task_four/glitchy_counter_4 (FF)
  Source Clock:      clk_in rising
  Destination Clock: clk_in rising

  Data Path: task_four/glitchy_counter_3 to task_four/glitchy_counter_4
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     FDR:C->Q              4   0.361   0.707  task_four/glitchy_counter_3
(task_four/glitchy_counter_3)
     LUT6:I0->O            2   0.097   0.299
task_four/Maccum_glitchy_counter_xor<6>111
(task_four/Maccum_glitchy_counter_xor<6>11)
     LUT4:I3->O            1   0.097   0.000
task_four/Maccum_glitchy_counter_xor<4>11 (Result<4>)
     FDR:D                     0.008          task_four/glitchy_counter_4
    ----------------------------------------
    Total                      1.570ns (0.563ns logic, 1.007ns route)
                                       (35.9% logic, 64.1% route)

=========================================================================
```

```
Timing constraint: Default OFFSET IN BEFORE for Clock 'clk_in'
  Total number of paths / destination ports: 27 / 27
--------------------------------------------------------------------------------
Offset:              0.884ns (Levels of Logic = 2)
  Source:            rst (PAD)
  Destination:       task_one/clk_div_2 (FF)
  Destination Clock: clk_in rising

  Data Path: rst to task_one/clk_div_2
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     IBUF:I->O            24   0.001   0.382  rst_IBUF (rst_IBUF)
     INV:I->O              4   0.113   0.293  task_four/clk2/rst_inv1_INV_0
(task_four/clk2/rst_inv)
     FDE:CE                    0.095          task_one/clk_div_16
    ----------------------------------------
    Total                      0.884ns (0.209ns logic, 0.675ns route)
                                       (23.6% logic, 76.4% route)

================================================================================
Timing constraint: Default OFFSET OUT AFTER for Clock 'clk_in'
  Total number of paths / destination ports: 19 / 14
--------------------------------------------------------------------------------
Offset:              1.445ns (Levels of Logic = 2)
  Source:            task_three/clk_neg_1 (FF)
  Destination:       clk_div_5 (PAD)
  Source Clock:      clk_in falling

  Data Path: task_three/clk_neg_1 to clk_div_5
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     FDC:C->Q              4   0.361   0.707  task_three/clk_neg_1
(task_three/clk_neg_1)
     LUT6:I0->O            1   0.097   0.279  task_three/clk_div_51
(clk_div_5_OBUF)
     OBUF:I->O                 0.000          clk_div_5_OBUF (clk_div_5)
    ----------------------------------------
    Total                      1.445ns (0.458ns logic, 0.987ns route)
                                       (31.7% logic, 68.3% route)

================================================================================

Cross Clock Domains Report:
--------------------------

Clock to Setup on destination clock clk_in
---------------+---------+---------+---------+---------+
               | Src:Rise| Src:Fall| Src:Rise| Src:Fall|
Source Clock   |Dest:Rise|Dest:Rise|Dest:Fall|Dest:Fall|
---------------+---------+---------+---------+---------+
clk_in         |    1.570|         |    0.991|         |
---------------+---------+---------+---------+---------+
```

```
================================================================================


Total REAL time to Xst completion: 8.00 secs
Total CPU time to Xst completion: 7.63 secs

-->


Total memory usage is 601332 kilobytes

Number of errors   :    0 (   0 filtered)
Number of warnings :   11 (   0 filtered)
Number of infos    :    6 (   0 filtered)
```

Map Report

```
Release 14.7 Map P.20131013 (lin64)
Xilinx Mapping Report File for Design 'clock_gen'

Design Information
------------------
Command Line   : map -intstyle ise -p xc7a100t-csg324-3 -w -logic_opt off -ol
high -t 1 -xt 0 -register_duplication off -r 4 -mt off -ir off -pr off -lc off
-power off -o clock_gen_map.ncd clock_gen.ngd clock_gen.pcf
Target Device  : xc7a100t
Target Package : csg324
Target Speed   : -3
Mapper Version : artix7 -- $Revision: 1.55 $
Mapped Date    : Sat May  9 22:03:45 2020

Design Summary
--------------
Number of errors:      0
Number of warnings:   18
Slice Logic Utilization:
  Number of Slice Registers:                    28 out of 126,800    1%
    Number used as Flip Flops:                  28
    Number used as Latches:                      0
    Number used as Latch-thrus:                  0
    Number used as AND/OR logics:                0
  Number of Slice LUTs:                         22 out of  63,400    1%
    Number used as logic:                       22 out of  63,400    1%
      Number using O6 output only:              15
      Number using O5 output only:               0
      Number using O5 and O6:                    7
      Number used as ROM:                        0
    Number used as Memory:                       0 out of  19,000    0%
    Number used exclusively as route-thrus:      0

Slice Logic Distribution:
  Number of occupied Slices:                    10 out of  15,850    1%
```

```
  Number of LUT Flip Flop pairs used:           25
    Number with an unused Flip Flop:             4 out of      25   16%
    Number with an unused LUT:                   3 out of      25   12%
    Number of fully used LUT-FF pairs:          18 out of      25   72%
    Number of unique control sets:               4
    Number of slice register sites lost
      to control set restrictions:              20 out of 126,800    1%

  A LUT Flip Flop pair for this architecture represents one LUT paired with
  one Flip Flop within a slice.  A control set is a unique combination of
  clock, reset, set, and enable signals for a registered element.
  The Slice Logic Distribution report is not meaningful if the design is
  over-mapped for a non-slice resource or if Placement fails.
  OVERMAPPING of BRAM resources should be ignored if the design is
  over-mapped for a non-BRAM resource or if placement fails.

IO Utilization:
  Number of bonded IOBs:                       16 out of     210    7%

Specific Feature Utilization:
  Number of RAMB36E1/FIFO36E1s:                 0 out of     135    0%
  Number of RAMB18E1/FIFO18E1s:                 0 out of     270    0%
  Number of BUFG/BUFGCTRLs:                     1 out of      32    3%
    Number used as BUFGs:                       1
    Number used as BUFGCTRLs:                   0
  Number of IDELAYE2/IDELAYE2_FINEDELAYs:       0 out of     300    0%
  Number of ILOGICE2/ILOGICE3/ISERDESE2s:       0 out of     300    0%
  Number of ODELAYE2/ODELAYE2_FINEDELAYs:       0
  Number of OLOGICE2/OLOGICE3/OSERDESE2s:       0 out of     300    0%
  Number of PHASER_IN/PHASER_IN_PHYs:           0 out of      24    0%
  Number of PHASER_OUT/PHASER_OUT_PHYs:         0 out of      24    0%
  Number of BSCANs:                             0 out of       4    0%
  Number of BUFHCEs:                            0 out of      96    0%
  Number of BUFRs:                              0 out of      24    0%
  Number of CAPTUREs:                           0 out of       1    0%
  Number of DNA_PORTs:                          0 out of       1    0%
  Number of DSP48E1s:                           0 out of     240    0%
  Number of EFUSE_USRs:                         0 out of       1    0%
  Number of FRAME_ECCs:                         0 out of       1    0%
  Number of IBUFDS_GTE2s:                       0 out of       4    0%
  Number of ICAPs:                              0 out of       2    0%
  Number of IDELAYCTRLs:                        0 out of       6    0%
  Number of IN_FIFOs:                           0 out of      24    0%
  Number of MMCME2_ADVs:                        0 out of       6    0%
  Number of OUT_FIFOs:                          0 out of      24    0%
  Number of PCIE_2_1s:                          0 out of       1    0%
  Number of PHASER_REFs:                        0 out of       6    0%
  Number of PHY_CONTROLs:                       0 out of       6    0%
  Number of PLLE2_ADVs:                         0 out of       6    0%
  Number of STARTUPs:                           0 out of       1    0%
  Number of XADCs:                              0 out of       1    0%

Average Fanout of Non-Clock Nets:               3.72
```

```
Peak Memory Usage:  1300 MB
Total REAL time to MAP completion:  31 secs
Total CPU time to MAP completion:   29 secs

Table of Contents
-----------------
Section 1 - Errors
Section 2 - Warnings
Section 3 - Informational
Section 4 - Removed Logic Summary
Section 5 - Removed Logic
Section 6 - IOB Properties
Section 7 - RPMs
Section 8 - Guide Report
Section 9 - Area Group and Partition Summary
Section 10 - Timing Report
Section 11 - Configuration String Information
Section 12 - Control Set Information
Section 13 - Utilization by Hierarchy

Section 1 - Errors
------------------

Section 2 - Warnings
--------------------
WARNING:LIT:701 - PAD symbol "clk_in" has an undefined IOSTANDARD.
WARNING:LIT:702 - PAD symbol "clk_in" is not constrained (LOC) to a specific
   location.
WARNING:PhysDesignRules:2452 - The IOB clk_div_16 is either not constrained
   (LOC) to a specific location and/or has an undefined I/O Standard
   (IOSTANDARD). This condition may seriously affect the device and will be an
   error in bitstream creation. It should be corrected by properly specifying
   the pin location and I/O Standard.
WARNING:PhysDesignRules:2452 - The IOB clk_div_28 is either not constrained
   (LOC) to a specific location and/or has an undefined I/O Standard
   (IOSTANDARD). This condition may seriously affect the device and will be an
   error in bitstream creation. It should be corrected by properly specifying
   the pin location and I/O Standard.
WARNING:PhysDesignRules:2452 - The IOB glitchy_counter<0> is either not
   constrained (LOC) to a specific location and/or has an undefined I/O
Standard
   (IOSTANDARD). This condition may seriously affect the device and will be an
   error in bitstream creation. It should be corrected by properly specifying
   the pin location and I/O Standard.
WARNING:PhysDesignRules:2452 - The IOB glitchy_counter<1> is either not
   constrained (LOC) to a specific location and/or has an undefined I/O
Standard
   (IOSTANDARD). This condition may seriously affect the device and will be an
   error in bitstream creation. It should be corrected by properly specifying
   the pin location and I/O Standard.
WARNING:PhysDesignRules:2452 - The IOB glitchy_counter<2> is either not
   constrained (LOC) to a specific location and/or has an undefined I/O
Standard
   (IOSTANDARD). This condition may seriously affect the device and will be an
```

```
   error in bitstream creation. It should be corrected by properly specifying
   the pin location and I/O Standard.
WARNING:PhysDesignRules:2452 - The IOB glitchy_counter<3> is either not
   constrained (LOC) to a specific location and/or has an undefined I/O
Standard
   (IOSTANDARD). This condition may seriously affect the device and will be an
   error in bitstream creation. It should be corrected by properly specifying
   the pin location and I/O Standard.
WARNING:PhysDesignRules:2452 - The IOB glitchy_counter<4> is either not
   constrained (LOC) to a specific location and/or has an undefined I/O
Standard
   (IOSTANDARD). This condition may seriously affect the device and will be an
   error in bitstream creation. It should be corrected by properly specifying
   the pin location and I/O Standard.
WARNING:PhysDesignRules:2452 - The IOB glitchy_counter<5> is either not
   constrained (LOC) to a specific location and/or has an undefined I/O
Standard
   (IOSTANDARD). This condition may seriously affect the device and will be an
   error in bitstream creation. It should be corrected by properly specifying
   the pin location and I/O Standard.
WARNING:PhysDesignRules:2452 - The IOB glitchy_counter<6> is either not
   constrained (LOC) to a specific location and/or has an undefined I/O
Standard
   (IOSTANDARD). This condition may seriously affect the device and will be an
   error in bitstream creation. It should be corrected by properly specifying
   the pin location and I/O Standard.
WARNING:PhysDesignRules:2452 - The IOB glitchy_counter<7> is either not
   constrained (LOC) to a specific location and/or has an undefined I/O
Standard
   (IOSTANDARD). This condition may seriously affect the device and will be an
   error in bitstream creation. It should be corrected by properly specifying
   the pin location and I/O Standard.
WARNING:PhysDesignRules:2452 - The IOB clk_in is either not constrained (LOC)
to
   a specific location and/or has an undefined I/O Standard (IOSTANDARD). This
   condition may seriously affect the device and will be an error in bitstream
   creation. It should be corrected by properly specifying the pin location and
   I/O Standard.
WARNING:PhysDesignRules:2452 - The IOB clk_div_2 is either not constrained
(LOC)
   to a specific location and/or has an undefined I/O Standard (IOSTANDARD).
   This condition may seriously affect the device and will be an error in
   bitstream creation. It should be corrected by properly specifying the pin
   location and I/O Standard.
WARNING:PhysDesignRules:2452 - The IOB clk_div_4 is either not constrained
(LOC)
   to a specific location and/or has an undefined I/O Standard (IOSTANDARD).
   This condition may seriously affect the device and will be an error in
   bitstream creation. It should be corrected by properly specifying the pin
   location and I/O Standard.
WARNING:PhysDesignRules:2452 - The IOB clk_div_5 is either not constrained
(LOC)
   to a specific location and/or has an undefined I/O Standard (IOSTANDARD).
   This condition may seriously affect the device and will be an error in
```

bitstream creation. It should be corrected by properly specifying the pin location and I/O Standard.
WARNING:PhysDesignRules:2452 - The IOB rst is either not constrained (LOC) to a specific location and/or has an undefined I/O Standard (IOSTANDARD). This condition may seriously affect the device and will be an error in bitstream creation. It should be corrected by properly specifying the pin location and I/O Standard.
WARNING:PhysDesignRules:2452 - The IOB clk_div_8 is either not constrained (LOC) to a specific location and/or has an undefined I/O Standard (IOSTANDARD). This condition may seriously affect the device and will be an error in bitstream creation. It should be corrected by properly specifying the pin location and I/O Standard.

Section 3 - Informational

INFO:LIT:244 - All of the single ended outputs in this design are using slew rate limited output drivers. The delay on speed critical single ended outputs can be dramatically reduced by designating them as fast outputs.
INFO:Pack:1716 - Initializing temperature to 85.000 Celsius. (default - Range: 0.000 to 85.000 Celsius)
INFO:Pack:1720 - Initializing voltage to 0.950 Volts. (default - Range: 0.950 to 1.050 Volts)
INFO:Map:215 - The Interim Design Summary has been generated in the MAP Report (.mrp).
INFO:Pack:1650 - Map created a placed design.

Section 4 - Removed Logic Summary

Section 5 - Removed Logic

To enable printing of redundant blocks removed and signals merged, set the detailed map report option and rerun map.

Section 6 - IOB Properties

IOB Name	Type	Direction	IO Standard	Diff	Drive	Slew	Reg (s)	Resistor	IOB Term	Strength	Rate		Delay
clk_div_2	IOB	OUTPUT	LVCMOS18		12	SLOW							
clk_div_4	IOB	OUTPUT	LVCMOS18		12	SLOW							

| clk_div_5 | IOB | OUTPUT | LVCMOS18
| 12 | SLOW | | | |
| clk_div_8 | IOB | OUTPUT | LVCMOS18
| 12 | SLOW | | | |
| clk_div_16 | IOB | OUTPUT | LVCMOS18
| 12 | SLOW | | | |
| clk_div_28 | IOB | OUTPUT | LVCMOS18
| 12 | SLOW | | | |
| clk_in | IOB | INPUT | LVCMOS18
| | | | | |
| glitchy_counter<0> | IOB | OUTPUT | LVCMOS18
| 12 | SLOW | | | |
| glitchy_counter<1> | IOB | OUTPUT | LVCMOS18
| 12 | SLOW | | | |
| glitchy_counter<2> | IOB | OUTPUT | LVCMOS18
| 12 | SLOW | | | |
| glitchy_counter<3> | IOB | OUTPUT | LVCMOS18
| 12 | SLOW | | | |
| glitchy_counter<4> | IOB | OUTPUT | LVCMOS18
| 12 | SLOW | | | |
| glitchy_counter<5> | IOB | OUTPUT | LVCMOS18
| 12 | SLOW | | | |
| glitchy_counter<6> | IOB | OUTPUT | LVCMOS18
| 12 | SLOW | | | |
| glitchy_counter<7> | IOB | OUTPUT | LVCMOS18
| 12 | SLOW | | | |
| rst | IOB | INPUT | LVCMOS18
| | | | | |
+---
---+

Section 7 - RPMs

Section 8 - Guide Report

Guide not run on this design.

Section 9 - Area Group and Partition Summary

Partition Implementation Status

  No Partitions were found in this design.

Area Group Information

  No area groups were found in this design.

```
Section 10 - Timing Report
--------------------------
A logic-level (pre-route) timing report can be generated by using Xilinx static
timing analysis tools, Timing Analyzer (GUI) or TRCE (command line), with the
mapped NCD and PCF files. Please note that this timing report will be generated
using estimated delay information. For accurate numbers, please generate a
timing report with the post Place and Route NCD file.

For more information about the Timing Analyzer, consult the Xilinx Timing
Analyzer Reference Manual; for more information about TRCE, consult the Xilinx
Command Line Tools User Guide "TRACE" chapter.

Section 11 - Configuration String Details
-----------------------------------------
Use the "-detail" map option to print out Configuration Strings

Section 12 - Control Set Information
------------------------------------
Use the "-detail" map option to print out Control Set Information.

Section 13 - Utilization by Hierarchy
-------------------------------------
Use the "-detail" map option to print out the Utilization by Hierarchy section.
```

While there are a few warnings that are generated and seen in the synthesis report and map report, I could not figure out what they meant or how to fix them, however they did not affect the performance of any of the tasks. Therefore, I believe this is alright.