Lab 2: Clock Design Methodology

Ethan Ngo

TA: Ananya Ravikumar (Section 3)

May 5, 2021

Contents

1 Introduction **3**

2 Module Description **4**

3 Test-bench Simulation **7**

4 Reports **8**

4.1 ISE Design Overview Summary Report 8

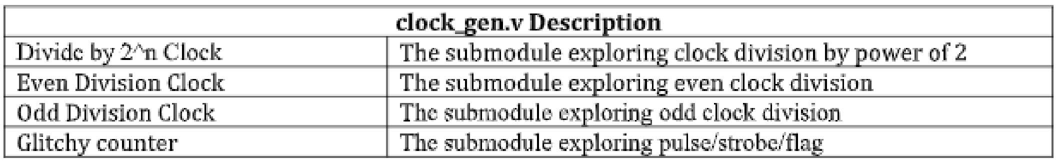
4.2 Synthesis Report 9

4.3 Map Report 23

5 Conclusion **29**

1 Introduction

The focus of the Clock Design Methodology lab is for students to become familiar with the Xilinx ISE software and utilize it to learn the basic concept behind clocking a system and how to generate them from a system clock. Clocks are useful for synchronously sending data. Taking a clock and reset signal, where the clock signal alternates at a specified frequency, we are to produce certain signals that varied in frequency and duty cycles, and even performed even and odd division. For this project, I created a top module of clock_gen and had a multitude of sub-modules to produce different signals to compare side-by-side. Below is a brief summary of the modules necessary:



clock_gen.v Description	
Divide by 2^n Clock	The submodule exploring clock division by power of 2
Even Division Clock	The submodule exploring even clock division
Odd Division Clock	The submodule exploring odd clock division
Glitchy counter	The submodule exploring pulse/strobe/flag

Figure 1: clock_gen Module Descriptions

2 Module Description

For the design of the various clocks, I adopted the module layout provided in the lab specifications, implementing these categories of clocks: Clock Divider by Powers of 2, Even Division Clock Using Counters, Odd Division Clock Using Counters, and Glitchy Counter With Pulse/Strobes. All of the modules take in a clock and reset signal, and all of them output some different type of signal.

1. **Category 1: Clock Divider by Powers of 2**

   This category dealt with division by powers of 2. Using one submodule clock_div_two, it output four clock signals: clock_div_2, clock_div_4, clock_div_8, and clock_div_16. Using the specs as a hint, I simply took the output Q[3:0] from the clock, that was incremented every time the clock was high, and used the 0th, 1st, 2nd, and 3rd most significant bits for each clock signal, respectively.

   In Verilog, I implemented this using an always block that was sensitive to the positive edge of the input clock. As mentioned above, I would increment Q and by 1. If rst was high, then it would be set to 0.

2. **Category 2: Even Division By Counters** For this category, we specifically deal with divison by 32 and 26, using sub-modules clock_div_thirty_two and clock_div_twenty_six. For division by 32, it was very similar to Category 1, however, since we only have a 32-bit value, it would overflow, and we dealt with that accordingly, reversing the clock signal on the 16th positive edge of the clock. Using the same logic, we would change the "overflow" value from 15 to 12, because with division by 26, the 13th positive edge would mean inverting the clock signal, so

going from 0-12 would be 13 clock signal alternations.

In Verilog, I implemented it as explained above. Using an always block, we would first reset if the rst signal was high. If not, we would check if Q == 4'b1111/4'b1100 to see if we had overflow, for division by 32 and 26, respectively. If we did, we would invert the clock signal and increment by 1. If not, we would just simply increment by 1.

3. **Category 3: Odd Division Using Counters**

   For this category, we handled division by 3 and 5 with sub-modules clock_div_thirty_three and clock_div_five. For 3, I used the examples shown in lecture to implement the counter portion. This counter count[1:0] would enable us to change the clock signal as necessary to achieve the required 33% duty cycle. Whenever the counter was 0, we would set the clock to high, leaving it low for 1 and 2. Combining the positive edge and negative edge of these signals using a logical or, we would get the 33% duty cycle, which was also explained in lecture.

   For division by 5, I had to implement a different strategy to effectively achieve what was required in the specs. I used a rotation method with a counter count[4:0] since that is 5 total bits we need. Specifically, using the counter, I would set the two most significant bit to be the value of the clock signal, and as the input clock changed, we would rotate the bits and the two most significant bit would change with these rotations. When the two signals are combined using a logical or, just like we did previously, we would get the 50% duty cycle required in the specs.

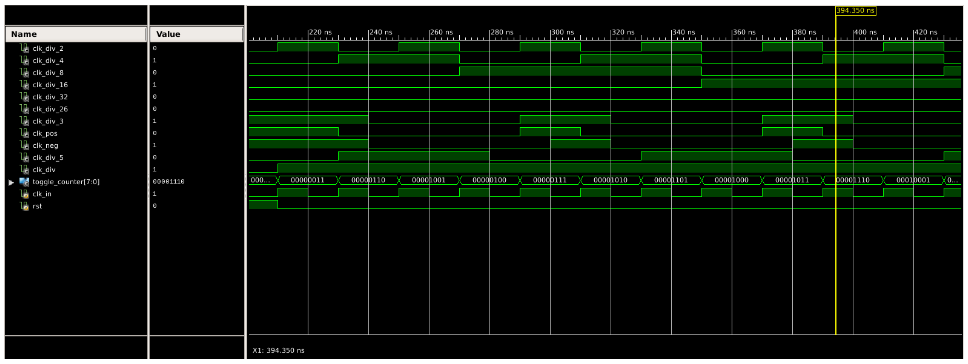
4. **Glitchy Counter With Pulse/Strobes**

For this category, our sub-module clock_pulse would handle division by 100 was very similar to above with 5, just the counter is count[99:0], for the necessary 100 bits. To maintain the 1% duty cycle, we would only need to set the most significant bit to be the clock signal. To create the 50% duty cycle division by 200 signal, we would invert the division by 100 signal whenever it reached 100 cycles, because this would have the cycle on for 100 cycles and off the 100 cycles, giving us a 50% duty cycle.

For the sub-module clock_strobe we would have a 4-strobe pulse on a 8-bit counter. When the most significant bit Q[3] is 1, we subtract by 5. Otherwise we add by 3. These operations are done on a counter toggle_counter[7:0], which would keep track of output.

3 Test-bench Simulation

To test my modules, I tested my main module clock_gen and compared the resulting waveforms to the specs. Below are snippets of the waveforms from the test bench:



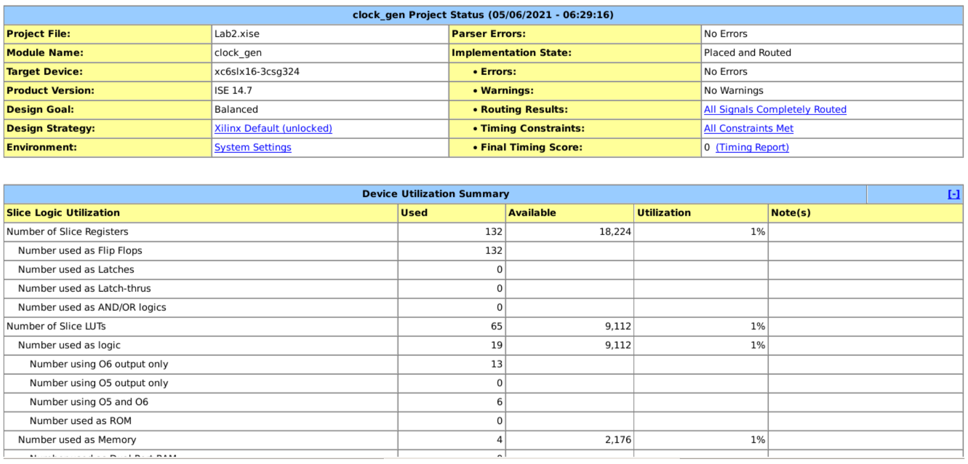
Name	Value
clk_div_2	0
clk_div_4	1
clk_div_8	0
clk_div_16	1
clk_div_32	0
clk_div_26	0
clk_div_3	1
clk_pos	0
clk_neg	1
clk_div_5	0
clk_div	1
toggle_counter[7:0]	00001110
clk_in	1
rst	0

Figure 2: clock_gen Simulation Waveforms Snippet

A bug I found was that while testing my modules, I wanted to give a short burst of clock input. Namely, turn the input clock on and off a few times and seeing what the output of the waveforms would be. However, for most of the waveforms, I got no response. It wasn't until I added a steady frequency for the input clock using an always block that the waveforms would respond correctly.

4 Reports

4.1 ISE Design Overview Summary Report



clock_gen Project Status (05/06/2021 - 06:29:16)			
Project File:	Lab2.xise	**Parser Errors:**	No Errors
Module Name:	clock_gen	**Implementation State:**	Placed and Routed
Target Device:	xc6slx16-3csg324	**• Errors:**	No Errors
Product Version:	ISE 14.7	**• Warnings:**	No Warnings
Design Goal:	Balanced	**• Routing Results:**	All Signals Completely Routed
Design Strategy:	Xilinx Default (unlocked)	**• Timing Constraints:**	All Constraints Met
Environment:	System Settings	**• Final Timing Score:**	0 (Timing Report)

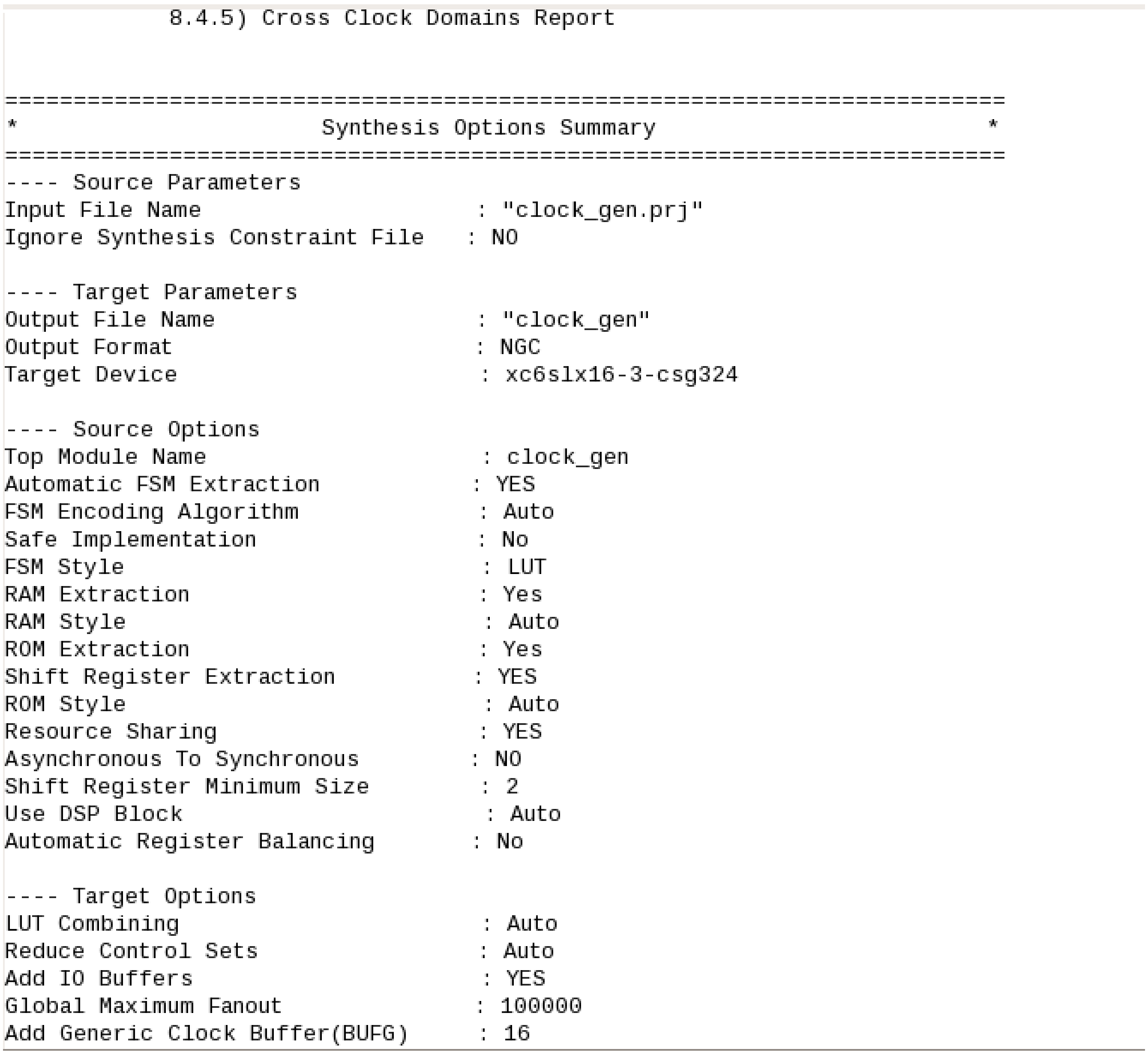
Device Utilization Summary				[-]
Slice Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Registers	132	18,224	1%	
Number used as Flip Flops	132			
Number used as Latches	0			
Number used as Latch-thrus	0			
Number used as AND/OR logics	0			
Number of Slice LUTs	65	9,112	1%	
Number used as logic	19	9,112	1%	
Number using O6 output only	13			
Number using O5 output only	0			
Number using O5 and O6	6			
Number used as ROM	0			
Number used as Memory	4	2,176	1%	

Figure 3: Design Summary

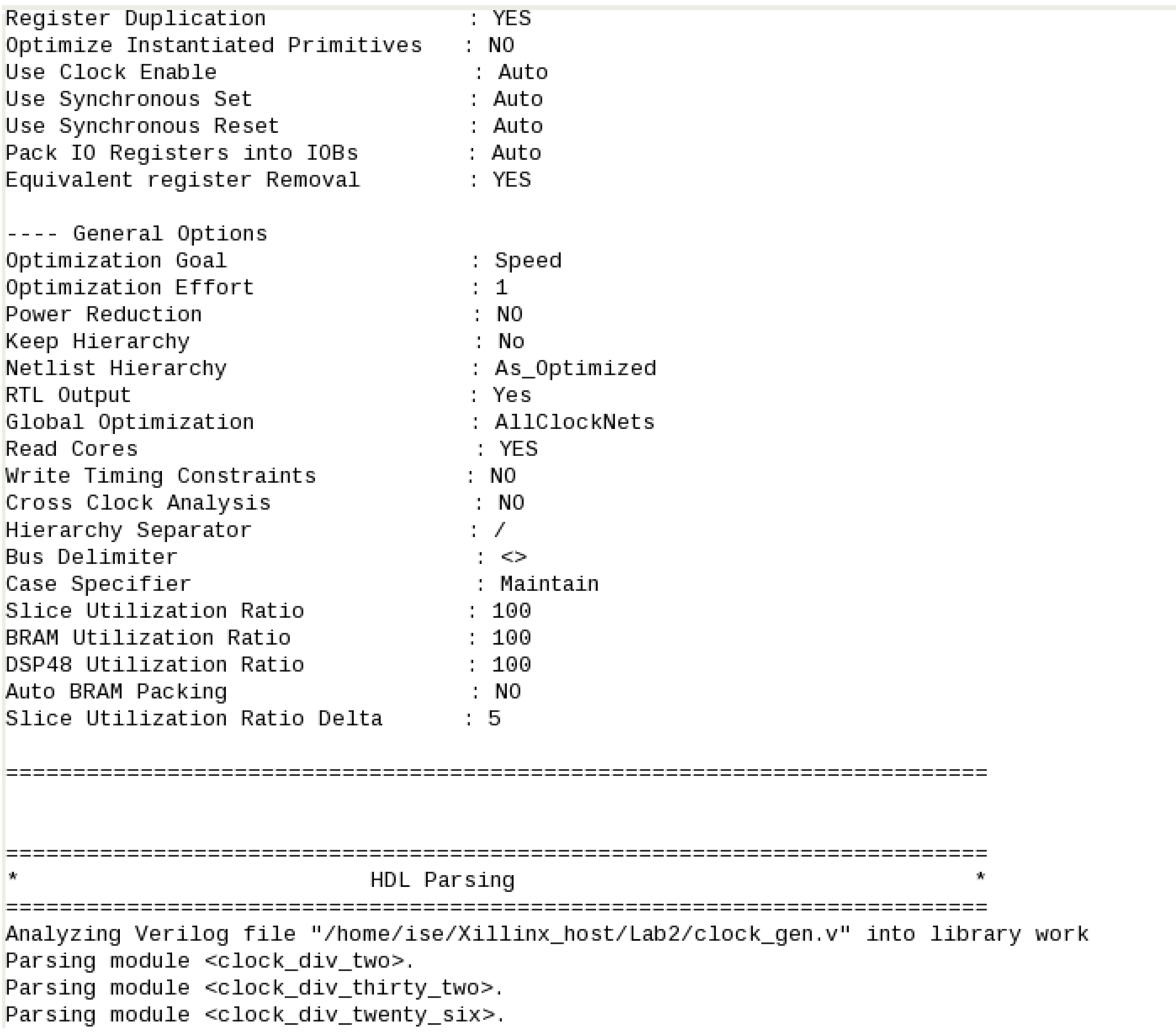
Above is the ISE Design Overview Summary Report. As indicated, there are no errors or warnings.

4.2 Synthesis Report

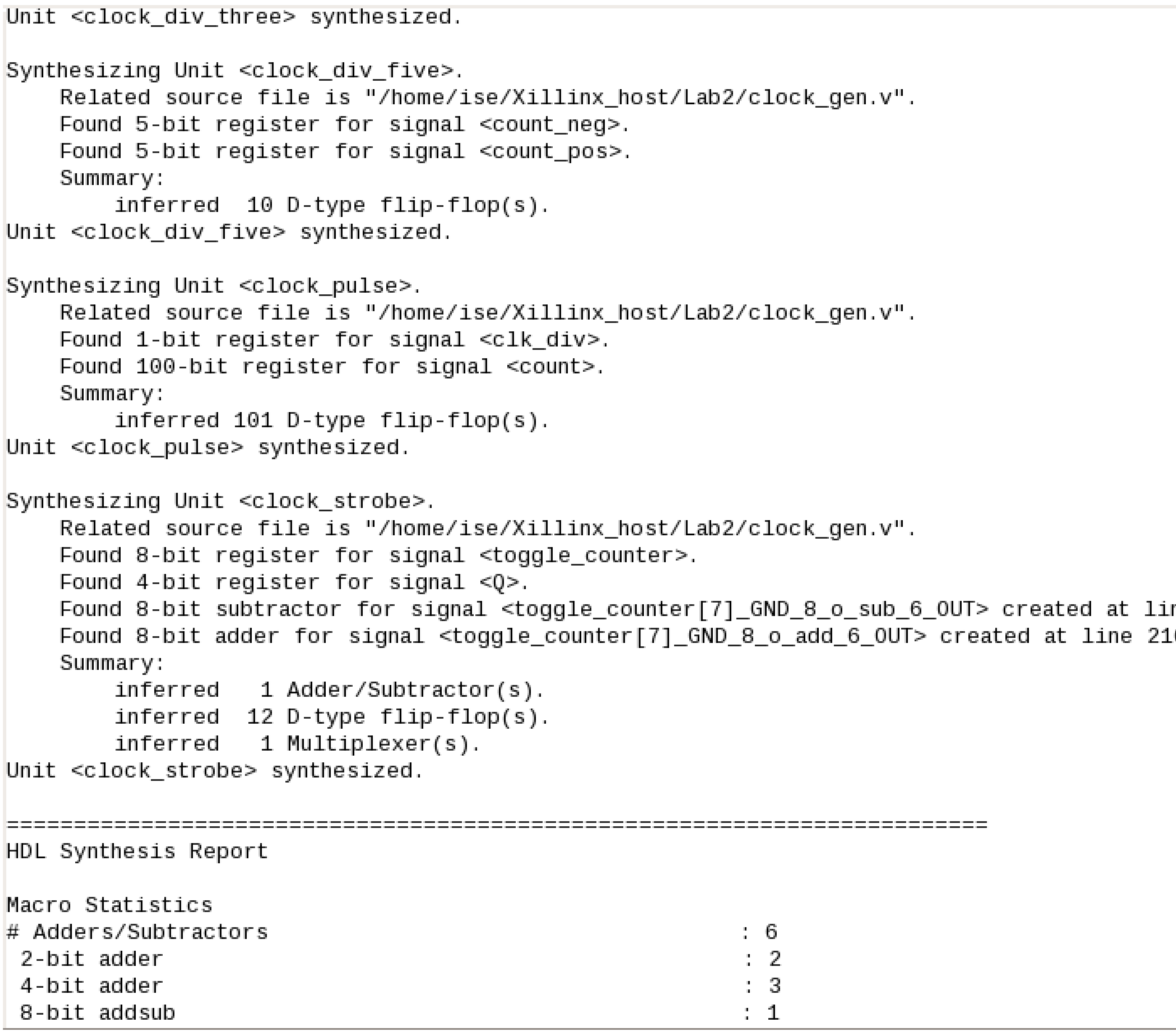
```
Release 14.7 - xst P.20131013 (lin64)
Copyright (c) 1995-2013 Xilinx, Inc.  All rights reserved.
-->
Parameter TMPDIR set to xst/projnav.tmp


Total REAL time to Xst completion: 0.00 secs
Total CPU time to Xst completion: 0.19 secs

-->
Parameter xsthdpdir set to xst


Total REAL time to Xst completion: 0.00 secs
Total CPU time to Xst completion: 0.19 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
            8.4.1) Clock Information
            8.4.2) Asynchronous Control Signals Information
            8.4.3) Timing Summary
            8.4.4) Timing Details
```



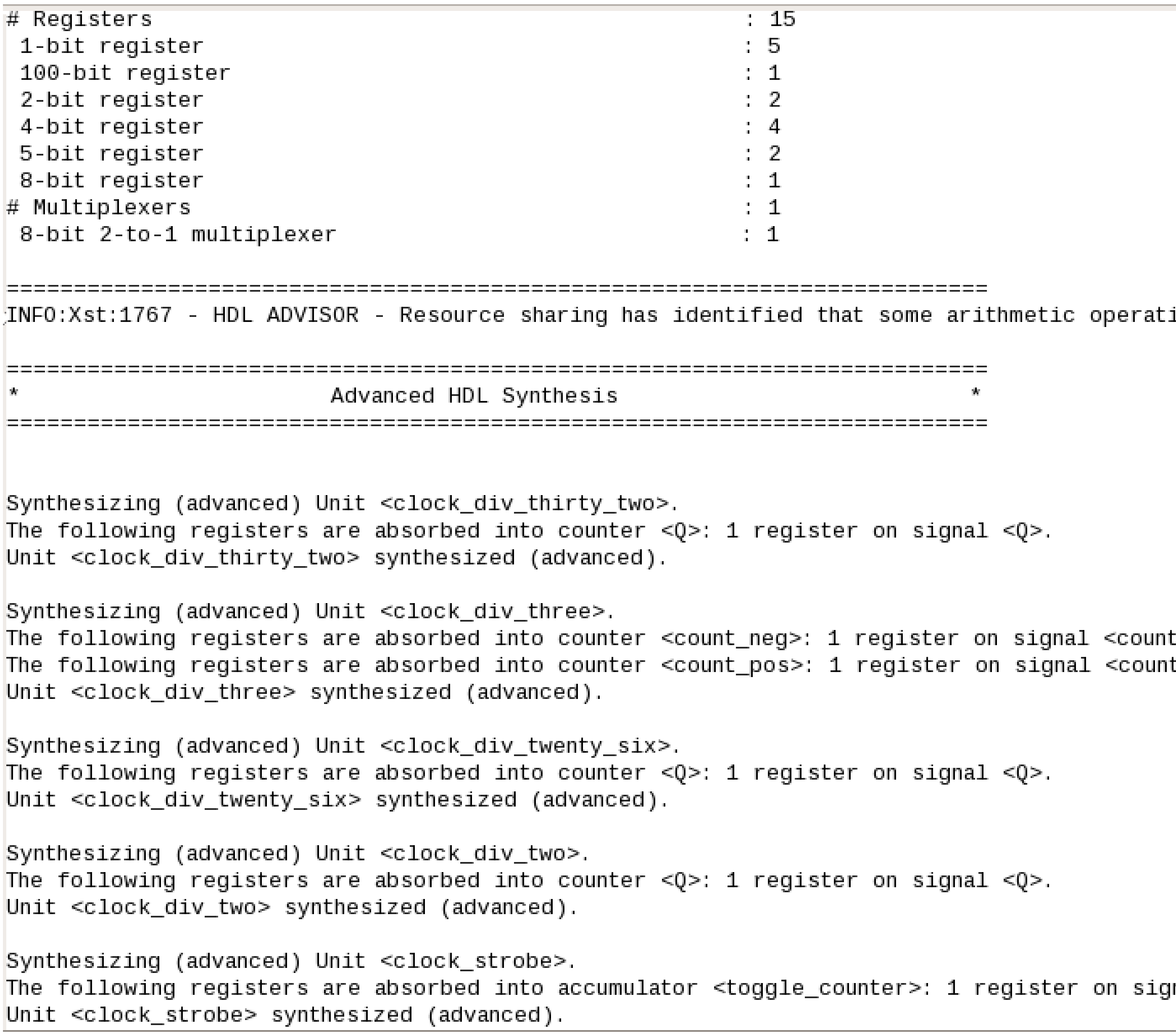
```
          8.4.5) Cross Clock Domains Report


=========================================================================
*                      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                      : xc6slx16-3-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)     : 16
```



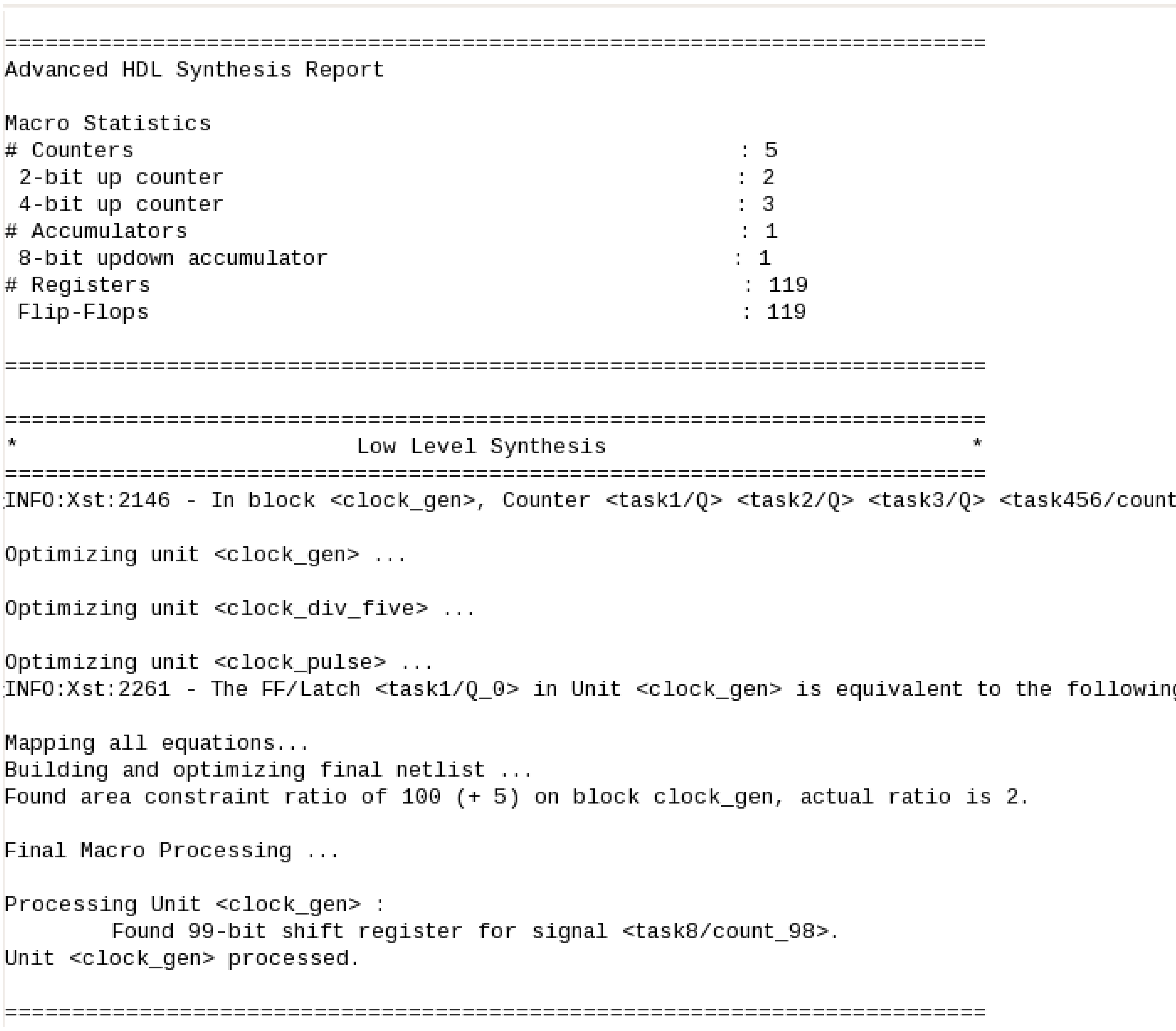
```
Register Duplication                : YES
Optimize Instantiated Primitives    : NO
Use Clock Enable                    : Auto
Use Synchronous Set                 : Auto
Use Synchronous Reset               : Auto
Pack IO Registers into IOBs         : Auto
Equivalent register Removal         : YES

---- 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/Xillinx_host/Lab2/clock_gen.v" into library work
Parsing module <clock_div_two>.
Parsing module <clock_div_thirty_two>.
Parsing module <clock_div_twenty_six>.
Parsing module <clock_div_three>.
Parsing module <clock_div_five>.
Parsing module <clock_pulse>.
Parsing module <clock_strobe>.
Parsing module <clock_gen>.

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

Elaborating module <clock_gen>.

Elaborating module <clock_div_two>.

Elaborating module <clock_div_thirty_two>.

Elaborating module <clock_div_twenty_six>.

Elaborating module <clock_div_three>.

Elaborating module <clock_div_five>.

Elaborating module <clock_pulse>.

Elaborating module <clock_strobe>.

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

Synthesizing Unit <clock_gen>.
    Related source file is "/home/ise/Xillinx_host/Lab2/clock_gen.v".
    Summary:
        no macro.
Unit <clock_gen> synthesized.

Synthesizing Unit <clock_div_two>.
    Related source file is "/home/ise/Xillinx_host/Lab2/clock_gen.v".
    Found 4-bit register for signal <Q>.
    Found 4-bit adder for signal <Q[3]_GND_2_o_add_1_OUT> created at line 33.
    Summary:
        inferred   1 Adder/Subtractor(s).
        inferred   4 D-type flip-flop(s).
Unit <clock_div_two> synthesized.

Synthesizing Unit <clock_div_thirty_two>.
    Related source file is "/home/ise/Xillinx_host/Lab2/clock_gen.v".
    Found 1-bit register for signal <clk_div_32>.
    Found 4-bit register for signal <Q>.
    Found 4-bit adder for signal <Q[3]_GND_3_o_add_3_OUT> created at line 62.
    Summary:
        inferred   1 Adder/Subtractor(s).
        inferred   5 D-type flip-flop(s).
Unit <clock_div_thirty_two> synthesized.

Synthesizing Unit <clock_div_twenty_six>.
    Related source file is "/home/ise/Xillinx_host/Lab2/clock_gen.v".
    Found 1-bit register for signal <clk_div_26>.
    Found 4-bit register for signal <Q>.
    Found 4-bit adder for signal <Q[3]_GND_4_o_add_3_OUT> created at line 86.
    Summary:
        inferred   1 Adder/Subtractor(s).
        inferred   5 D-type flip-flop(s).
Unit <clock_div_twenty_six> synthesized.

Synthesizing Unit <clock_div_three>.
    Related source file is "/home/ise/Xillinx_host/Lab2/clock_gen.v".
    Found 1-bit register for signal <clk_pos>.
    Found 2-bit register for signal <count_neg>.
    Found 1-bit register for signal <clk_neg>.
    Found 2-bit register for signal <count_pos>.
    Found 2-bit adder for signal <count_pos[1]_GND_5_o_add_1_OUT> created at line 104.
    Found 2-bit adder for signal <count_neg[1]_GND_5_o_add_7_OUT> created at line 121.
    Summary:
        inferred   2 Adder/Subtractor(s).
        inferred   6 D-type flip-flop(s).
```



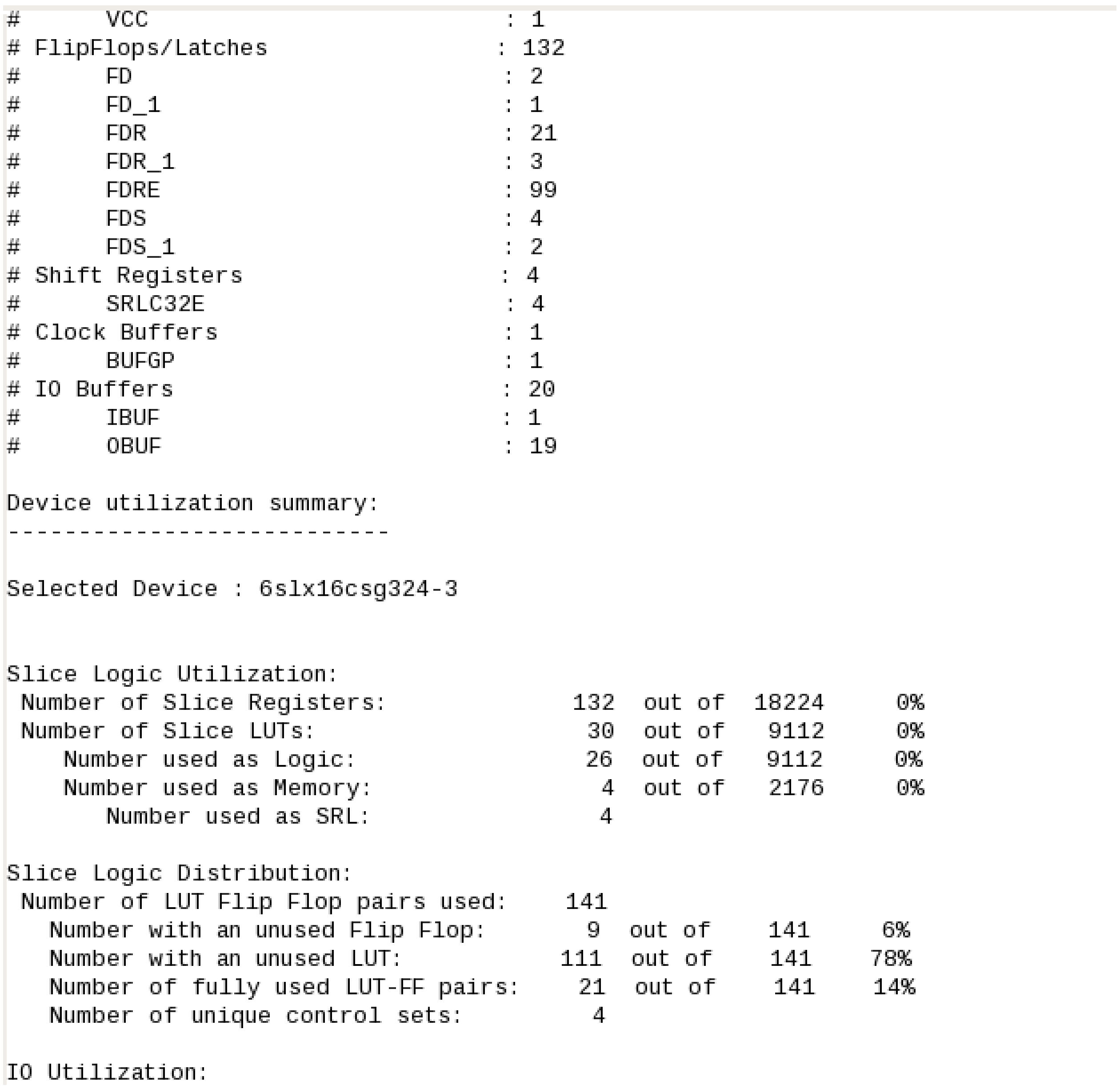
```
Unit <clock_div_three> synthesized.

Synthesizing Unit <clock_div_five>.
    Related source file is "/home/ise/Xillinx_host/Lab2/clock_gen.v".
    Found 5-bit register for signal <count_neg>.
    Found 5-bit register for signal <count_pos>.
    Summary:
        inferred  10 D-type flip-flop(s).
Unit <clock_div_five> synthesized.

Synthesizing Unit <clock_pulse>.
    Related source file is "/home/ise/Xillinx_host/Lab2/clock_gen.v".
    Found 1-bit register for signal <clk_div>.
    Found 100-bit register for signal <count>.
    Summary:
        inferred 101 D-type flip-flop(s).
Unit <clock_pulse> synthesized.

Synthesizing Unit <clock_strobe>.
    Related source file is "/home/ise/Xillinx_host/Lab2/clock_gen.v".
    Found 8-bit register for signal <toggle_counter>.
    Found 4-bit register for signal <Q>.
    Found 8-bit subtractor for signal <toggle_counter[7]_GND_8_o_sub_6_OUT> created at lin
    Found 8-bit adder for signal <toggle_counter[7]_GND_8_o_add_6_OUT> created at line 21
    Summary:
        inferred   1 Adder/Subtractor(s).
        inferred  12 D-type flip-flop(s).
        inferred   1 Multiplexer(s).
Unit <clock_strobe> synthesized.

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

Macro Statistics
# Adders/Subtractors                                   : 6
 2-bit adder                                           : 2
 4-bit adder                                           : 3
 8-bit addsub                                          : 1
```



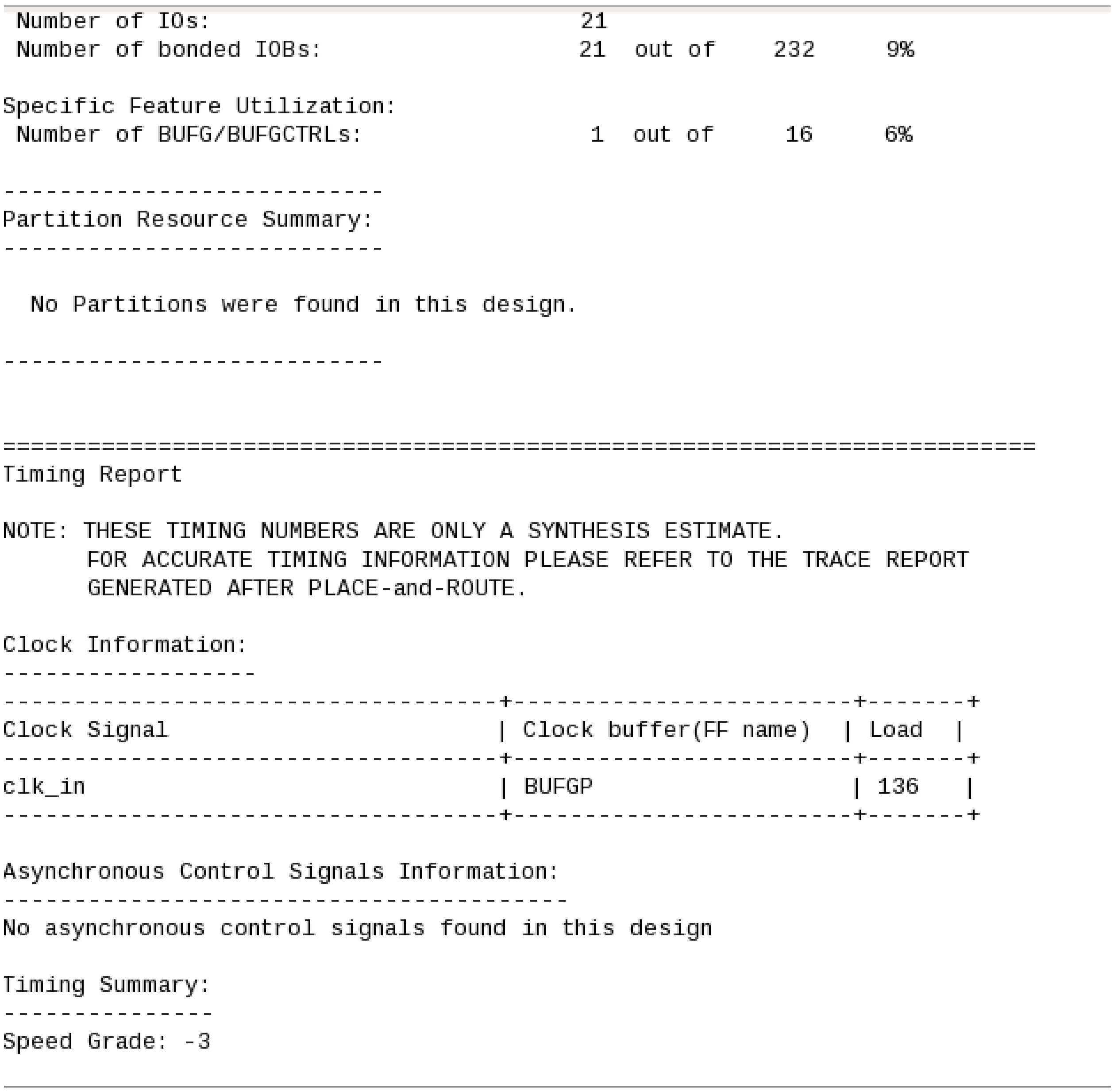
```
# Registers                                              : 15
 1-bit register                                          : 5
 100-bit register                                        : 1
 2-bit register                                          : 2
 4-bit register                                          : 4
 5-bit register                                          : 2
 8-bit register                                          : 1
# Multiplexers                                           : 1
 8-bit 2-to-1 multiplexer                                : 1

=========================================================================
INFO:Xst:1767 - HDL ADVISOR - Resource sharing has identified that some arithmetic operati

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


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

Synthesizing (advanced) Unit <clock_div_three>.
The following registers are absorbed into counter <count_neg>: 1 register on signal <count
The following registers are absorbed into counter <count_pos>: 1 register on signal <count
Unit <clock_div_three> synthesized (advanced).

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

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

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



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

Macro Statistics
# Counters                                                        : 5
 2-bit up counter                                                 : 2
 4-bit up counter                                                 : 3
# Accumulators                                                    : 1
 8-bit updown accumulator                                         : 1
# Registers                                                       : 119
 Flip-Flops                                                       : 119

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

================================================================================
*                          Low Level Synthesis                                 *
================================================================================
INFO:Xst:2146 - In block <clock_gen>, Counter <task1/Q> <task2/Q> <task3/Q> <task456/count

Optimizing unit <clock_gen> ...

Optimizing unit <clock_div_five> ...

Optimizing unit <clock_pulse> ...
INFO:Xst:2261 - The FF/Latch <task1/Q_0> in Unit <clock_gen> is equivalent to the following

Mapping all equations...
Building and optimizing final netlist ...
Found area constraint ratio of 100 (+ 5) on block clock_gen, actual ratio is 2.

Final Macro Processing ...

Processing Unit <clock_gen> :
        Found 99-bit shift register for signal <task8/count_98>.
Unit <clock_gen> processed.

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

Macro Statistics
# Registers                                            : 33
 Flip-Flops                                            : 33
# Shift Registers                                      : 1
 99-bit shift register                                 : 1

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

=========================================================================
*                            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                             : 28
#      GND                         : 1
#      INV                         : 3
#      LUT2                        : 8
#      LUT3                        : 4
#      LUT4                        : 2
#      LUT5                        : 6
#      LUT6                        : 3
```

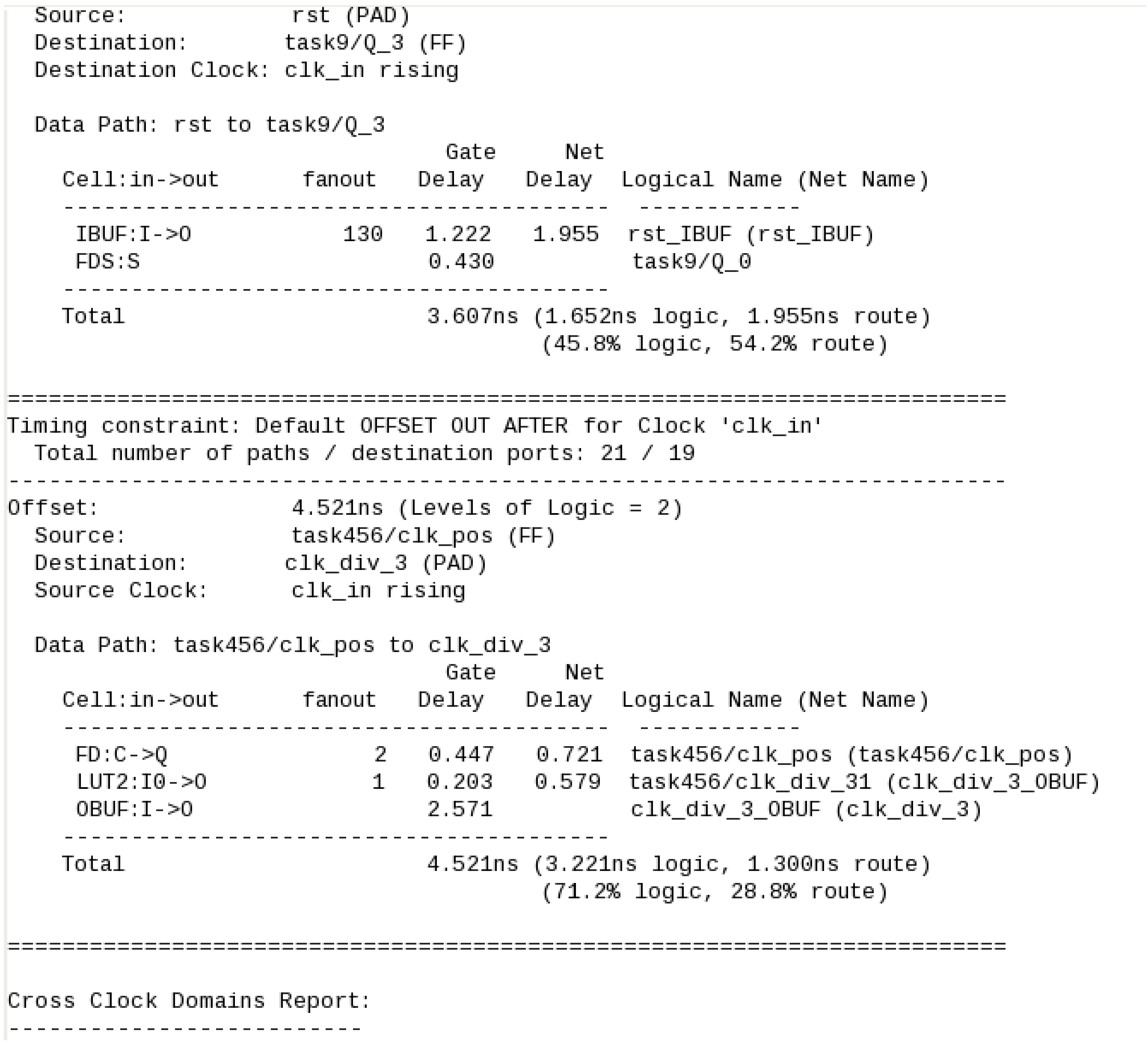


```
#      VCC                          : 1
# FlipFlops/Latches                 : 132
#      FD                           : 2
#      FD_1                         : 1
#      FDR                          : 21
#      FDR_1                        : 3
#      FDRE                         : 99
#      FDS                          : 4
#      FDS_1                        : 2
# Shift Registers                   : 4
#      SRLC32E                      : 4
# Clock Buffers                     : 1
#      BUFGP                        : 1
# IO Buffers                        : 20
#      IBUF                         : 1
#      OBUF                         : 19

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

Selected Device : 6slx16csg324-3


Slice Logic Utilization:
 Number of Slice Registers:             132  out of  18224     0%
 Number of Slice LUTs:                   30  out of   9112     0%
    Number used as Logic:                26  out of   9112     0%
    Number used as Memory:                4  out of   2176     0%
       Number used as SRL:                4

Slice Logic Distribution:
 Number of LUT Flip Flop pairs used:    141
   Number with an unused Flip Flop:       9  out of    141     6%
   Number with an unused LUT:           111  out of    141    78%
   Number of fully used LUT-FF pairs:    21  out of    141    14%
   Number of unique control sets:         4

IO Utilization:
```



```
 Number of IOs:                          21
 Number of bonded IOBs:                  21  out of    232     9%

Specific Feature Utilization:
 Number of BUFG/BUFGCTRLs:                1  out of     16     6%

---------------------------
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                  | 136   |
-----------------------------------+------------------------+-------+

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

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

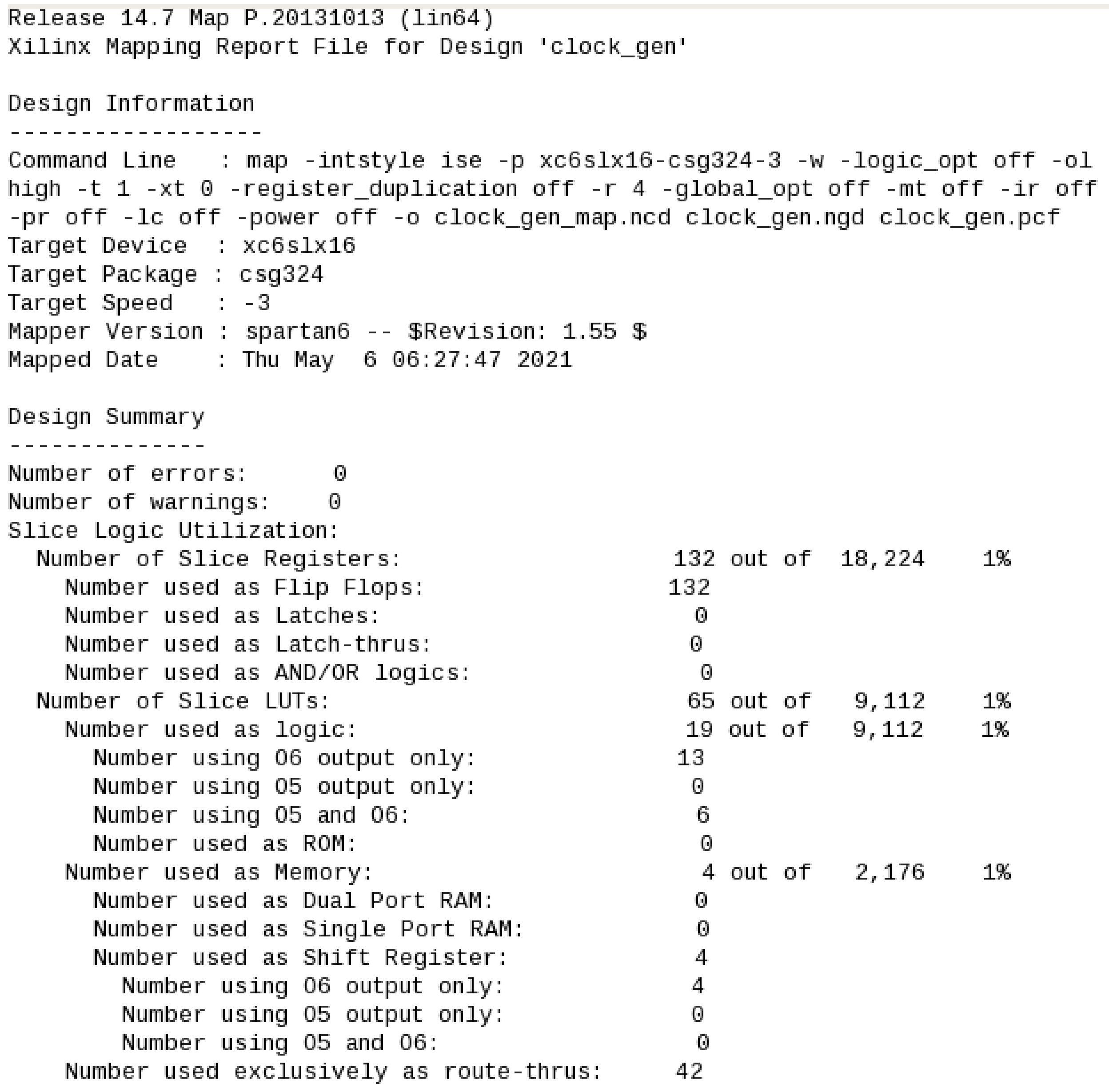
```
   Minimum period: 2.757ns (Maximum Frequency: 362.779MHz)
   Minimum input arrival time before clock: 3.607ns
   Maximum output required time after clock: 4.521ns
   Maximum combinational path delay: No path found

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

=========================================================================
Timing constraint: Default period analysis for Clock 'clk_in'
  Clock period: 2.757ns (frequency: 362.779MHz)
  Total number of paths / destination ports: 189 / 132
-------------------------------------------------------------------------
Delay:               2.757ns (Levels of Logic = 2)
  Source:            task9/toggle_counter_3 (FF)
  Destination:       task9/toggle_counter_7 (FF)
  Source Clock:      clk_in rising
  Destination Clock: clk_in rising

  Data Path: task9/toggle_counter_3 to task9/toggle_counter_7
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     FDR:C->Q              7   0.447   1.118  task9/toggle_counter_3 (task9/toggle_counte
     LUT5:I0->O            1   0.203   0.684  task9/Maccum_toggle_counter_xor<7>11_SW1 (N
     LUT6:I4->O            1   0.203   0.000  task9/Maccum_toggle_counter_xor<7>11 (Resul
     FDR:D                     0.102          task9/toggle_counter_7
    ----------------------------------------
    Total                      2.757ns (0.955ns logic, 1.802ns route)
                                       (34.6% logic, 65.4% route)

=========================================================================
Timing constraint: Default OFFSET IN BEFORE for Clock 'clk_in'
  Total number of paths / destination ports: 130 / 130
-------------------------------------------------------------------------
Offset:              3.607ns (Levels of Logic = 1)
```



```
  Source:             rst (PAD)
  Destination:        task9/Q_3 (FF)
  Destination Clock: clk_in rising

  Data Path: rst to task9/Q_3
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     IBUF:I->O           130   1.222   1.955  rst_IBUF (rst_IBUF)
     FDS:S                     0.430          task9/Q_0
    ----------------------------------------
    Total                      3.607ns (1.652ns logic, 1.955ns route)
                                       (45.8% logic, 54.2% route)

=========================================================================
Timing constraint: Default OFFSET OUT AFTER for Clock 'clk_in'
  Total number of paths / destination ports: 21 / 19
-------------------------------------------------------------------------
Offset:              4.521ns (Levels of Logic = 2)
  Source:            task456/clk_pos (FF)
  Destination:       clk_div_3 (PAD)
  Source Clock:      clk_in rising

  Data Path: task456/clk_pos to clk_div_3
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     FD:C->Q               2   0.447   0.721  task456/clk_pos (task456/clk_pos)
     LUT2:I0->O            1   0.203   0.579  task456/clk_div_31 (clk_div_3_OBUF)
     OBUF:I->O                 2.571          clk_div_3_OBUF (clk_div_3)
    ----------------------------------------
    Total                      4.521ns (3.221ns logic, 1.300ns route)
                                       (71.2% logic, 28.8% route)

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

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

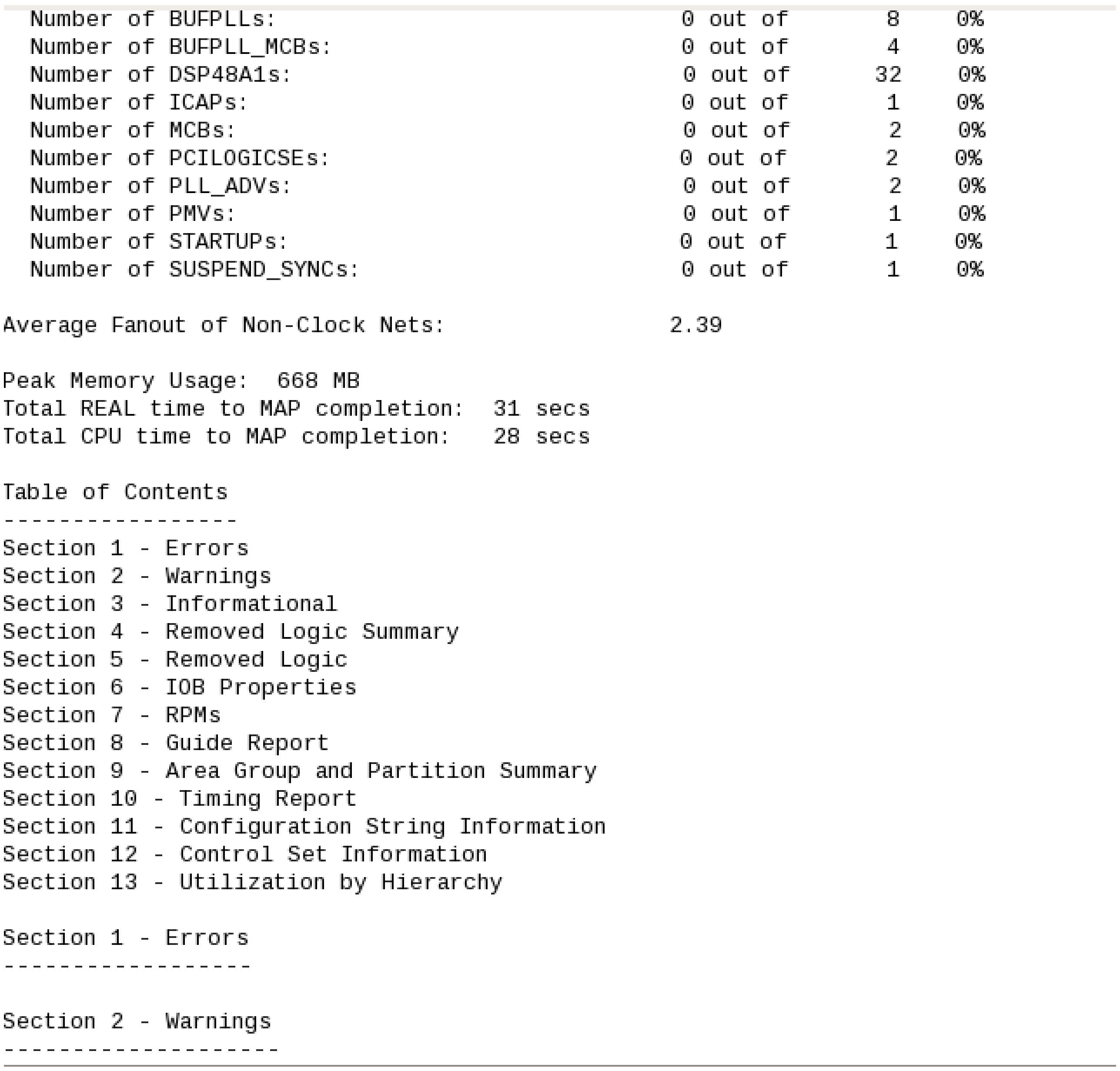
```
Clock to Setup on destination clock clk_in
---------------+---------+---------+---------+---------+
               | Src:Rise| Src:Fall| Src:Rise| Src:Fall|
Source Clock   |Dest:Rise|Dest:Rise|Dest:Fall|Dest:Fall|
---------------+---------+---------+---------+---------+
clk_in         |    2.757|         |    1.984|         |
---------------+---------+---------+---------+---------+

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


Total REAL time to Xst completion: 25.00 secs
Total CPU time to Xst completion: 20.00 secs

-->


Total memory usage is 386804 kilobytes

Number of errors   :    0 (   0 filtered)
Number of warnings :    0 (   0 filtered)
Number of infos    :    3 (   0 filtered)
```

4.3 Map Report

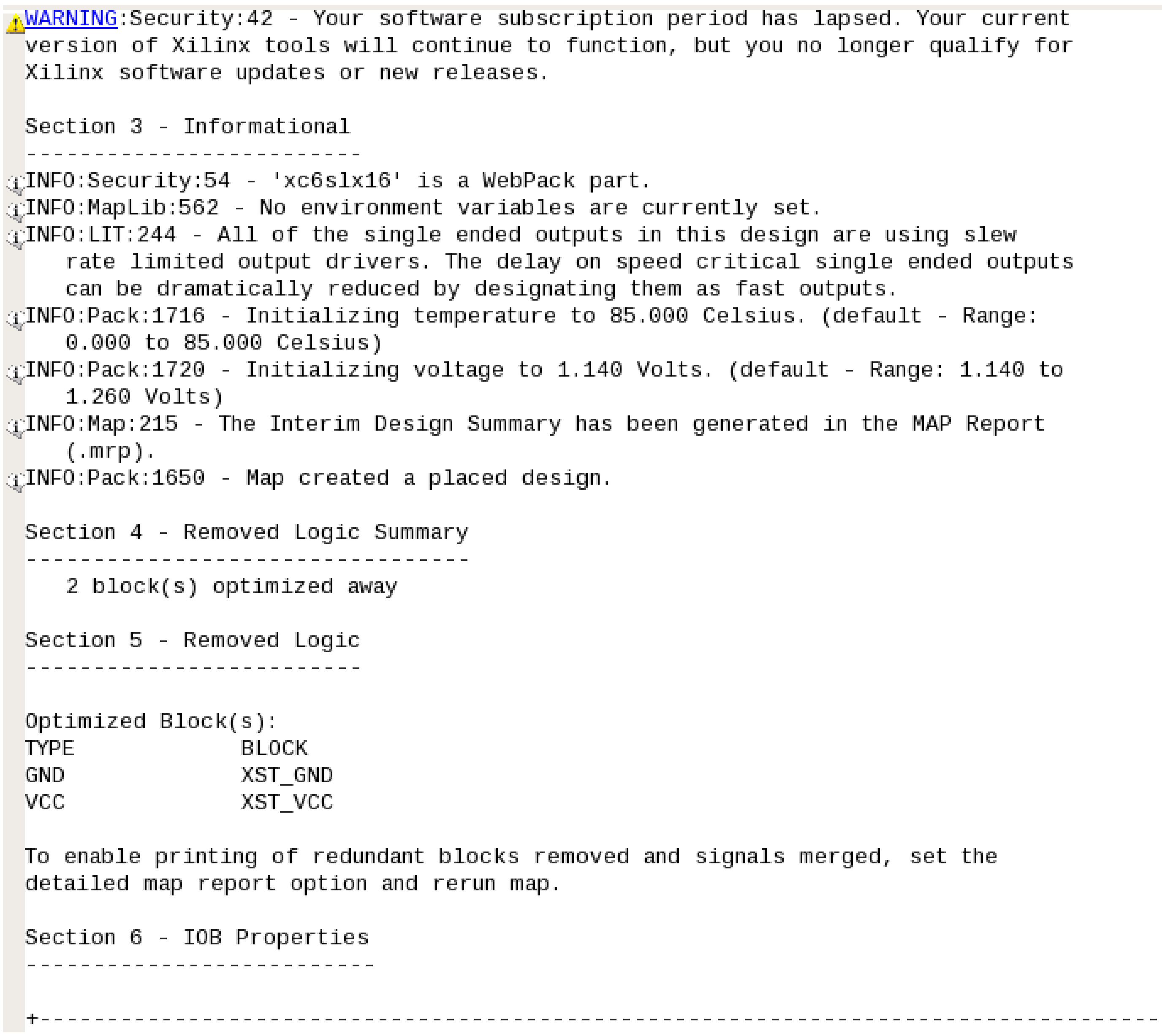


```
Release 14.7 Map P.20131013 (lin64)
Xilinx Mapping Report File for Design 'clock_gen'

Design Information
------------------
Command Line   : map -intstyle ise -p xc6slx16-csg324-3 -w -logic_opt off -ol
high -t 1 -xt 0 -register_duplication off -r 4 -global_opt off -mt off -ir off
-pr off -lc off -power off -o clock_gen_map.ncd clock_gen.ngd clock_gen.pcf
Target Device  : xc6slx16
Target Package : csg324
Target Speed   : -3
Mapper Version : spartan6 -- $Revision: 1.55 $
Mapped Date    : Thu May  6 06:27:47 2021

Design Summary
--------------
Number of errors:      0
Number of warnings:    0
Slice Logic Utilization:
  Number of Slice Registers:                   132 out of  18,224    1%
    Number used as Flip Flops:                 132
    Number used as Latches:                      0
    Number used as Latch-thrus:                  0
    Number used as AND/OR logics:                0
  Number of Slice LUTs:                         65 out of   9,112    1%
    Number used as logic:                       19 out of   9,112    1%
      Number using O6 output only:              13
      Number using O5 output only:               0
      Number using O5 and O6:                    6
      Number used as ROM:                        0
    Number used as Memory:                       4 out of   2,176    1%
      Number used as Dual Port RAM:              0
      Number used as Single Port RAM:            0
      Number used as Shift Register:             4
        Number using O6 output only:             4
        Number using O5 output only:             0
        Number using O5 and O6:                  0
    Number used exclusively as route-thrus:     42
```

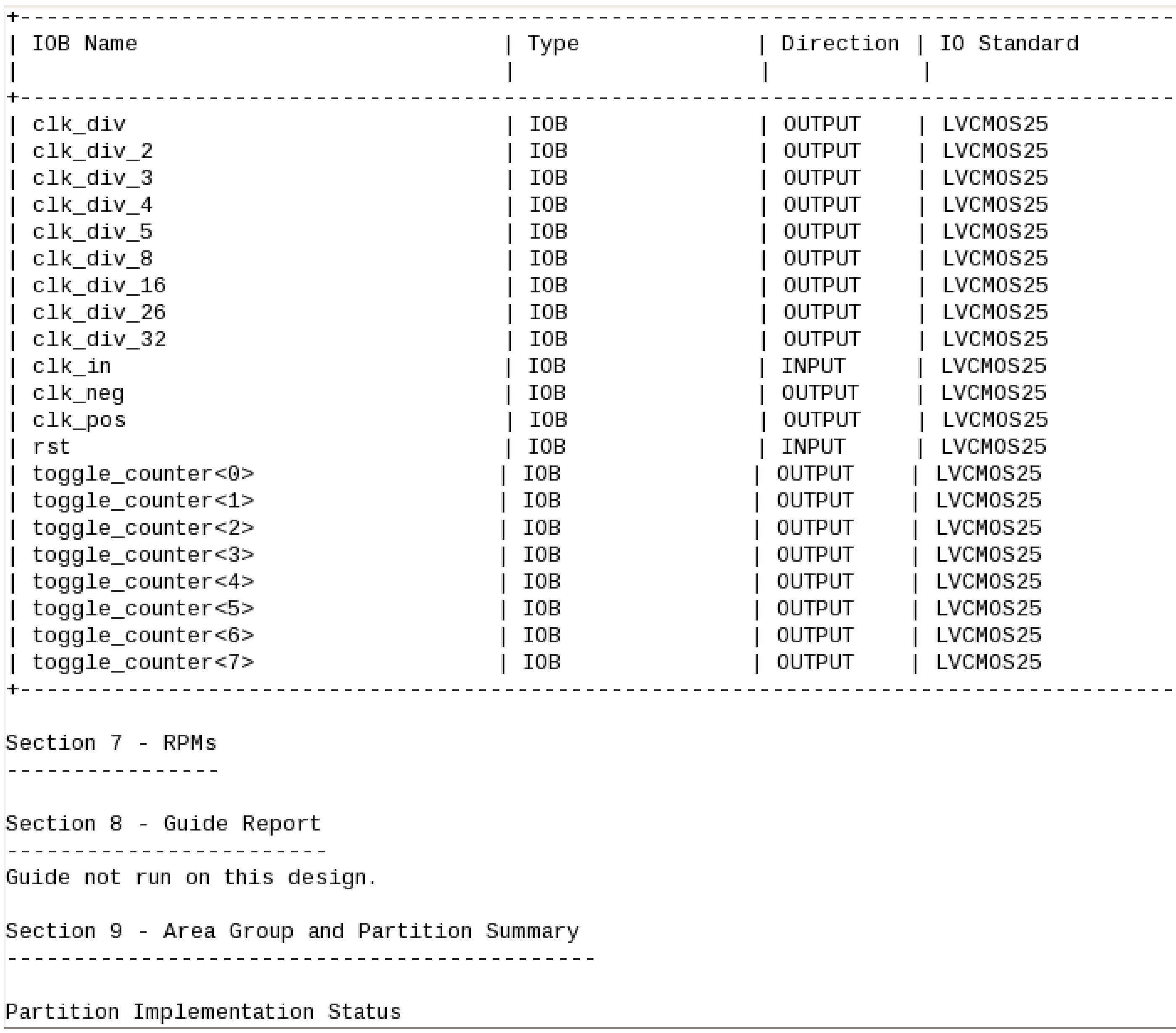
```
      Number with same-slice register load:     42
      Number with same-slice carry load:         0
      Number with other load:                    0

Slice Logic Distribution:
  Number of occupied Slices:                    27 out of   2,278    1%
  Number of MUXCYs used:                         0 out of   4,556    0%
  Number of LUT Flip Flop pairs used:           93
    Number with an unused Flip Flop:             7 out of      93    7%
    Number with an unused LUT:                  28 out of      93   30%
    Number of fully used LUT-FF pairs:          58 out of      93   62%
    Number of unique control sets:               5
    Number of slice register sites lost
      to control set restrictions:              24 out of  18,224    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.

IO Utilization:
  Number of bonded IOBs:                        21 out of     232    9%

Specific Feature Utilization:
  Number of RAMB16BWERs:                         0 out of      32    0%
  Number of RAMB8BWERs:                          0 out of      64    0%
  Number of BUFIO2/BUFIO2_2CLKs:                 0 out of      32    0%
  Number of BUFIO2FB/BUFIO2FB_2CLKs:             0 out of      32    0%
  Number of BUFG/BUFGMUXs:                       1 out of      16    6%
    Number used as BUFGs:                        1
    Number used as BUFGMUX:                      0
  Number of DCM/DCM_CLKGENs:                     0 out of       4    0%
  Number of ILOGIC2/ISERDES2s:                   0 out of     248    0%
  Number of IODELAY2/IODRP2/IODRP2_MCBs:         0 out of     248    0%
  Number of OLOGIC2/OSERDES2s:                   0 out of     248    0%
  Number of BSCANs:                              0 out of       4    0%
  Number of BUFHs:                               0 out of     128    0%
```



```
  Number of BUFPLLs:                             0 out of       8    0%
  Number of BUFPLL_MCBs:                         0 out of       4    0%
  Number of DSP48A1s:                            0 out of      32    0%
  Number of ICAPs:                               0 out of       1    0%
  Number of MCBs:                                0 out of       2    0%
  Number of PCILOGICSEs:                         0 out of       2    0%
  Number of PLL_ADVs:                            0 out of       2    0%
  Number of PMVs:                                0 out of       1    0%
  Number of STARTUPs:                            0 out of       1    0%
  Number of SUSPEND_SYNCs:                       0 out of       1    0%

Average Fanout of Non-Clock Nets:              2.39

Peak Memory Usage:  668 MB
Total REAL time to MAP completion:  31 secs
Total CPU time to MAP completion:   28 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:Security:42 - Your software subscription period has lapsed. Your current
version of Xilinx tools will continue to function, but you no longer qualify for
Xilinx software updates or new releases.

Section 3 - Informational
-------------------------
INFO:Security:54 - 'xc6slx16' is a WebPack part.
INFO:MapLib:562 - No environment variables are currently set.
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 1.140 Volts. (default - Range: 1.140 to
   1.260 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
---------------------------------
   2 block(s) optimized away

Section 5 - Removed Logic
-------------------------

Optimized Block(s):
TYPE            BLOCK
GND             XST_GND
VCC             XST_VCC

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
|                                    |                  |           |
+-----------------------------------------------------------------------------------------
| clk_div                            | IOB              | OUTPUT    | LVCMOS25
| clk_div_2                          | IOB              | OUTPUT    | LVCMOS25
| clk_div_3                          | IOB              | OUTPUT    | LVCMOS25
| clk_div_4                          | IOB              | OUTPUT    | LVCMOS25
| clk_div_5                          | IOB              | OUTPUT    | LVCMOS25
| clk_div_8                          | IOB              | OUTPUT    | LVCMOS25
| clk_div_16                         | IOB              | OUTPUT    | LVCMOS25
| clk_div_26                         | IOB              | OUTPUT    | LVCMOS25
| clk_div_32                         | IOB              | OUTPUT    | LVCMOS25
| clk_in                             | IOB              | INPUT     | LVCMOS25
| clk_neg                            | IOB              | OUTPUT    | LVCMOS25
| clk_pos                            | IOB              | OUTPUT    | LVCMOS25
| rst                                | IOB              | INPUT     | LVCMOS25
| toggle_counter<0>                  | IOB              | OUTPUT    | LVCMOS25
| toggle_counter<1>                  | IOB              | OUTPUT    | LVCMOS25
| toggle_counter<2>                  | IOB              | OUTPUT    | LVCMOS25
| toggle_counter<3>                  | IOB              | OUTPUT    | LVCMOS25
| toggle_counter<4>                  | IOB              | OUTPUT    | LVCMOS25
| toggle_counter<5>                  | IOB              | OUTPUT    | LVCMOS25
| toggle_counter<6>                  | IOB              | OUTPUT    | LVCMOS25
| toggle_counter<7>                  | IOB              | OUTPUT    | LVCMOS25
+-----------------------------------------------------------------------------------------

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.
```

5 Conclusion

In this lab, I designed a clock system that utilized clocks to create a variety of output signals. Using the knowledge from Lab 1 and the various tips and examples from the specifications and lecture, I was able to create these waveforms. I found it far easier to implement the even signals and the division by 3 signals.

Some difficulties I ran into was how to implement less straightforward signals, namely, everything past division by 3. In addition, I struggled to figure out how to create a new project and didn't realize how to do so, which create problems for me when I needed to test my code.