CSM152A-LAB 5

Lab 5

Yichen Lyu
UID: 004940413

May 24, 2020

1. INTRODUCTION

This lab will explore how to design, implement and test using Xilinx a parking meter finite state machine (FSM). It simulates the process of coins being added to the meter and meter displaying addition in time. The time is displayed in seconds, with a maximum value of 9999. The display of the time is done through Nexys 3 4-digit seven-segment LED.

2. DESIGN DESCRIPTION

The overall FSM will have three states: RESET, COUNT_DOWN, AND LESS_180. The state enters RESET when the input RST is high, or when the second count SECONDS become 0. In RESET, the four digits are four 0s and flashes at the rate of 1Hz and 50% duty cycle. If SECONDS is larger than 180, the state enters COUNT_DOWN, where SECONDS is reduced by 1 every second and the four digits display the decimal number for SECONDS, without flashing. If SECONDS is less or equal to 180, it enters LESS_180, where SECONDS counts down 1 per second, and the output four digits are the decimal representation of SECONDS, flashing with a period of 2 seconds and 50% duty cycle.

The overall design of the FSM is given in the graph below.

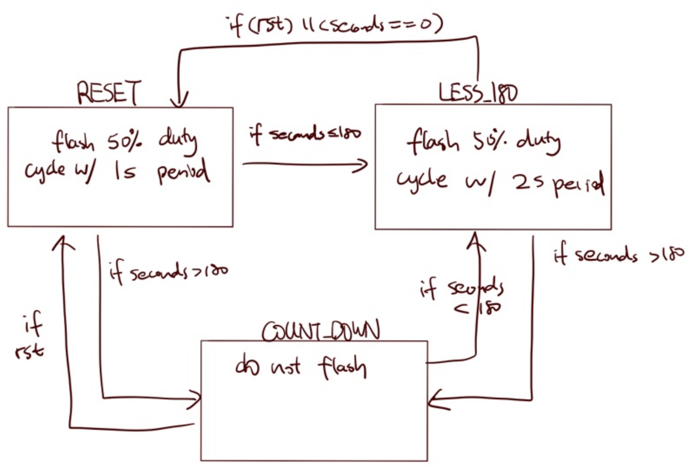

Figure 1: Overall FSM Design

There are several blocks I use during coding process to divide the problem down.

2.1 1Hz Clock Divider

The input clock CLK is 100Hz, so we first need to divide it into 1Hz for counting down the seconds. This is implemented by a 6-bit counter COUNT that counts from 0 up to 49. Then, in another always block, we flip the value of CLK_1Hz every time the counter counts to 49. If RST is high, CLK_1Hz is set to 0.

2.2 Refresh and Flashing

In the refresh block, the LED refreshes at the highest frequency possible: 200Hz. This block is triggered every posedge and negedge of the input clock. Every time, it will refresh one digit by changing the anodes and the cathodes. The total period of refreshing the whole LED is 50Hz because we have 4 digits to refresh in total. Upon the first refresh, we set the anode to 1000 because we want to display the first digit, and we talk in the first decimal value in SECONDS, which is stored in VAL1, and transform it into the 8-bit seven-segment representation in LED_SEG. Upon second refresh, we set the anode to 0100 and LED_SEG to the corresponding value in VAL2, and so on. After the four digits are represented, we go back to refreshing the first digit.

When dealing with flashing, we check the current state in CUR_STATE. If it’s refresh, we display the digits when 1HZ_CLK is high and do not display it when 1HZ_CLK is low with an if-statement by setting anode values all to 0. We do not do anything if the state is COUNT_DOWN. We use the value in VAL4 to determine whether to flash or not: if it’s even, we display. Otherwise, we set anodes to 0000.

2.3 SECONDS

We modify the value in SECONDS in another always block which is triggered by all the posedge of ADD and RST signals, as well as our 1Hz clock. Using if-statements, we add time to SECONDS depending on the type of ADD signals. Then we use if-statements to check the RST signals. Finally, if 1HZ_CLK_1% is high, it means that we enter this cycle because we detect a posedge of the 1% duty clock, and we should deduct 1 from SECONDS.

2.4 Value Assignment

This block is triggered by every input. It assigns current state depending on the value stored in SECONDS as described above and store the decimal value of seconds into VAL1, VAL2, VAL3, and VAL4. This is achieved by taking the remainder of SECONDS of 10 after division by 1, 10, 100, and 1000.

3. SCHEMATICS

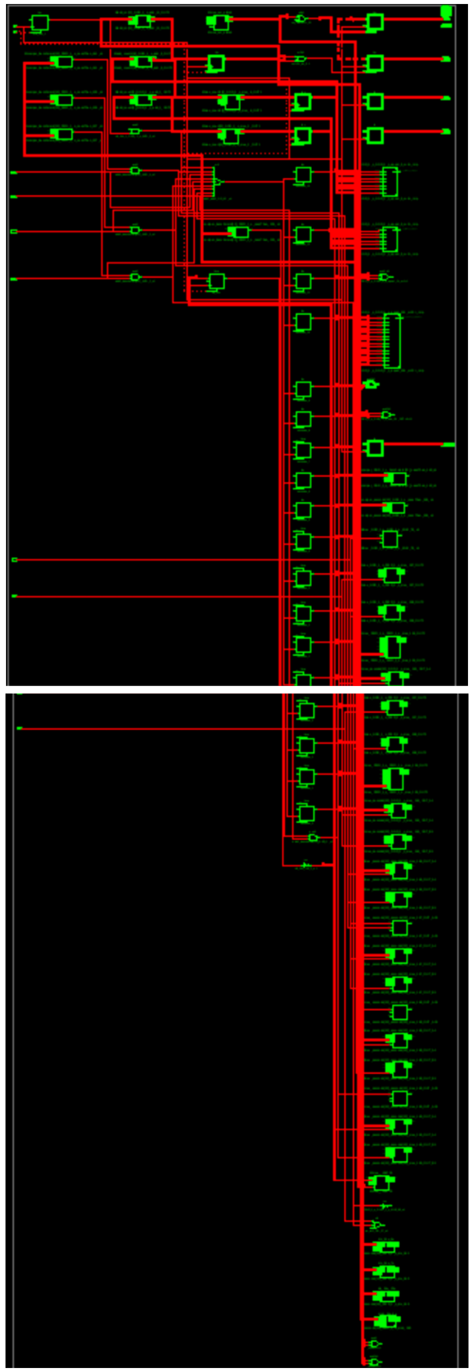

Figure 2: Auto Generated Schematics

The generated schematics is very complex. It would be clearer if I chose to implement the parking meter in different modules, but I put them all in one module. It is consisted with multiple MUXes because I'm constantly choosing the values based on inputs.

4. SIMULATION

4.1 Anode and Cathode Representation

The implementation of anodes follows the below graph. When a 1-bit anode is high, the digit it is representing is not shown on the LED. When it is low, the corresponding cathode value is represented. A1, A2, A3, and A4 represent the first, second, third, and the fourth digit, respectively. Upon each refresh, we set one anode low, following the order of A1,2,3,4. The refresh period is typically between 1ms to 16ms so human eyes see the constant display of all four digits. In this experiment, the fastest I can do is upon every input clock edge, so the rate is 200Hz.

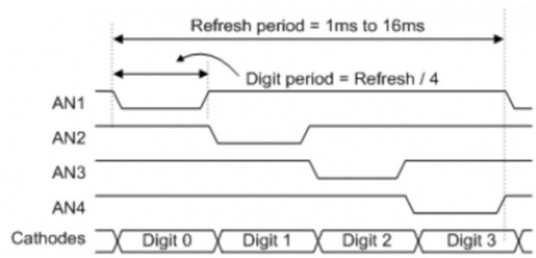

Figure 3: Anode Representation

For cathodes, according to the manuscript, we follow the below graph. CA-CG each represents an edge. I'm following the common anode notation, so when the edge is low, it is illuminated. The manuscript says we need to output with CA as the most significant bit and does not mention anything about DP, so my implementation is 7-bit, from CA-CG: 0 is 7'b0000001, 1 is 7'b1001111, 2 is 7'b0010010, and so on.

Figure 4: Cathode Representation

4.2 Refresh

The anodes refresh on every edge (1/2 of the input clk cycle), alternating to be low from a1 to a4. LED_SEG represents the digit corresponding to the low anode.

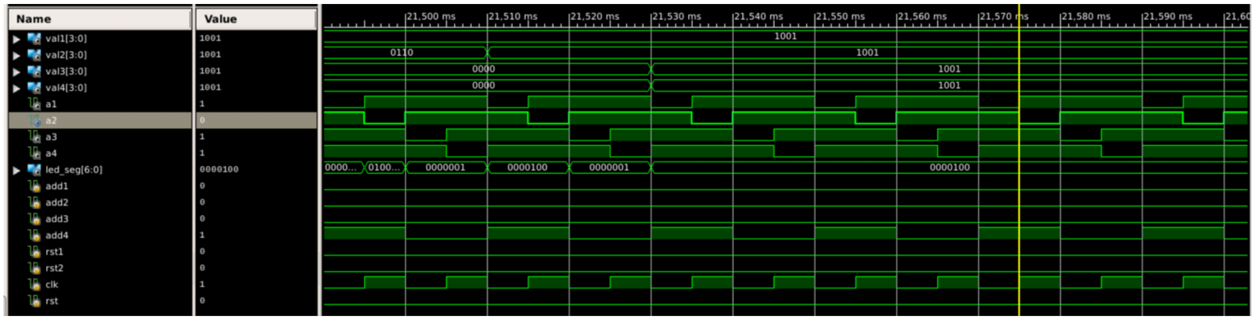

Figure 5: Refresh Waveform

4.3 Over 180

In the waveform below, SECONDS counts down from 9999, 9998… There is no flashing and, the anodes are always refreshing. VAL1, 2, and 3 remain 9, and VAL4 counts down from 9 to 8, 7... every second. The values in LED_SEG is changed based on which anode is low.

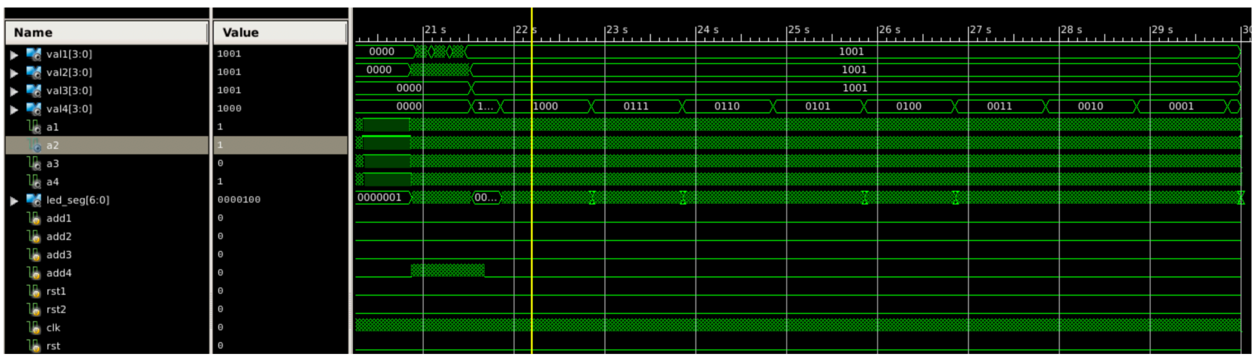

Figure 6: Normal Count Down Waveform

4.4 Less than 180

When the remaining time is less than 180s, the clock flashes with a period of 2s. When the displayed number is even, the digits are displayed; otherwise, no digit is displayed. This is achieved by setting all anodes to 1 when the number is odd and resume normal refreshing when it's even. The below simulation sets RST1 high, so it counts down from 15s. The digits are displayed when the number is even, as we can see that the anodes are alternatively low every other second. The value displayed in VAL1 and VAL2 are 0s, and VAL3 is initially 1 and then 0. VAL4 starts with 5 and then counts down. LED_SEG displays the seven-segment representation of the VAL whose anode is high, so it's constantly alternating.

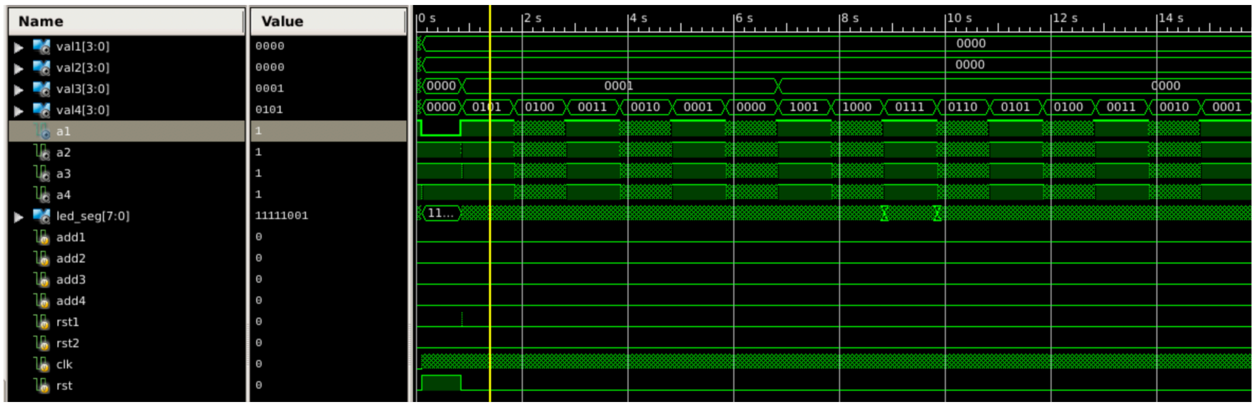

Figure 6: Less than 180s Waveforms

4.5 Reset

When SECONDS counts down to 0 or when RESET is set high, we enter the reset state, where the digits flash with a period of 1 second. As seen in the waveform, the anodes are refreshing for the first half of the second, and they stop displaying digits at the second half. The value of LED_SEG is the forever the seven-segment representation for 0 because we are flashing 0s.

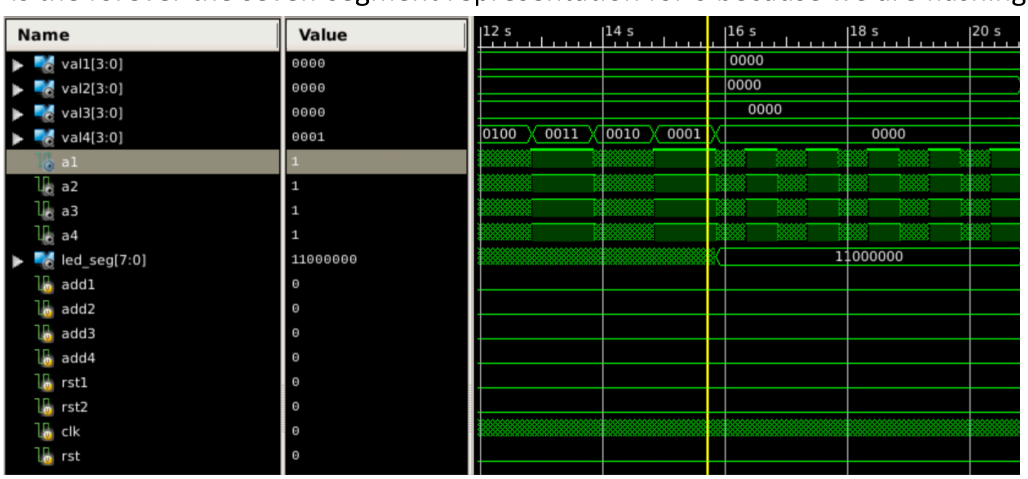

Figure 7: Reset Waveforms

4.6 Adding over 9999

After keep pressing ADD4, we reach the maximum of SECONDS: 9999. No matter how we keep pressing ADD4 after the maximum is reached, the values keep representing 9999 and count down from there.

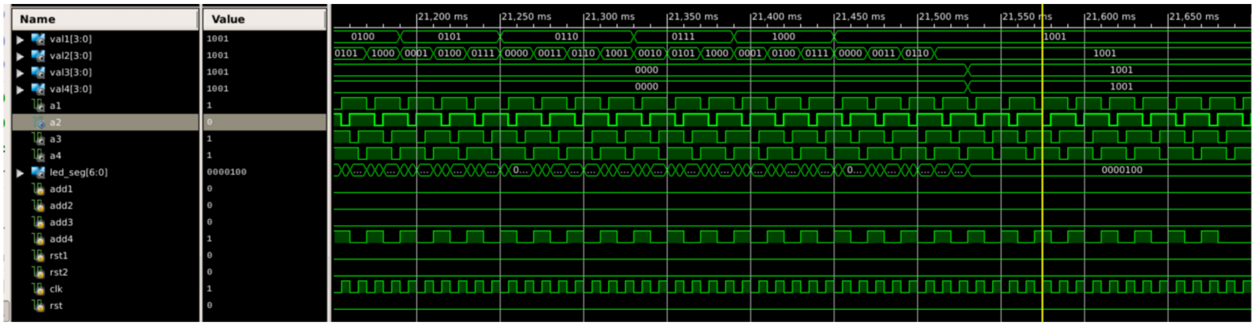

Figure 8: Over 9999s Waveform

5. CONCLUSION

This lab allows us to explore how to build a parking meter LED display using Nexys 3 seven-segment representation of anodes and cathodes in Xilinx ISE. It helps us learn how to utilize clock dividers and counters to real life problems. It also helps use learn how to use the concept for an FSM where we divide the problem into states to simplify implementation.

6. APPENDIX: DESIGN SUMMARY REPORT

6.1 Design Summary

parking_meter Project Status (06/01/2020 - 00:32:41)			
Project File:	lab5.xise	**Parser Errors:**	No Errors
Module Name:	parking_meter	**Implementation State:**	Placed and Routed
Target Device:	xc6slx16-3csg324	**• Errors:**	No Errors
Product Version:	ISE 14.7	**• Warnings:**	94 Warnings (79 new)
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	45	18,224	1%	
Number used as Flip Flops	27			
Number used as Latches	18			
Number used as Latch-thrus	0			
Number used as AND/OR logics	0			
Number of Slice LUTs	637	9,112	6%	
Number used as logic	635	9,112	6%	
Number using O6 output only	474			
Number using O5 output only	0			
Number using O5 and O6	161			
Number used as ROM	0			
Number used as Memory	0	2,176	0%	
Number used exclusively as route-thrus	2			
Number with same-slice register load	0			
Number with same-slice carry load	2			
Number with other load	0			
Number of occupied Slices	234	2,278	10%	
Number of MUXCYs used	240	4,556	5%	
Number of LUT Flip Flop pairs used	638			
Number with an unused Flip Flop	596	638	93%	
Number with an unused LUT	1	638	1%	
Number of fully used LUT-FF pairs	41	638	6%	
Number of unique control sets	7			
Number of slice register sites lost to control set restrictions	43	18,224	1%	
Number of bonded IOBs	63	232	27%	
Number of RAMB16BWERs	0	32	0%	
Number of RAMB8BWERs	0	64	0%	
Number of BUFIO2/BUFIO2_2CLKs	0	32	0%	
Number of BUFIO2FB/BUFIO2FB_2CLKs	0	32	0%	
Number of BUFG/BUFGMUXs	1	16	6%	
Number used as BUFGs	1			
Number used as BUFGMUX	0			
Number of DCM/DCM_CLKGENs	0	4	0%	
Number of ILOGIC2/ISERDES2s	0	248	0%	
Number of IODELAY2/IODRP2/IODRP2_MCBs	0	248	0%	
Number of OLOGIC2/OSERDES2s	0	248	0%	
Number of BSCANs	0	4	0%	
Number of BUFHs	0	128	0%	
Number of BUFPLLs	0	8	0%	
Number of BUFPLL_MCBs	0	4	0%	
Number of DSP48A1s	0	32	0%	
Number of ICAPs	0	1	0%	
Number of MCBs	0	2	0%	
Number of PCILOGICSEs	0	2	0%	

Number of PLL_ADVs	0	2	0%	
Number of PMVs	0	1	0%	
Number of STARTUPs	0	1	0%	
Number of SUSPEND_SYNCs	0	1	0%	
Average Fanout of Non-Clock Nets	4.31			

Performance Summary			[-]
Final Timing Score:	0 (Setup: 0, Hold: 0)	**Pinout Data:**	Pinout Report
Routing Results:	All Signals Completely Routed	**Clock Data:**	Clock Report
Timing Constraints:	All Constraints Met		

Detailed Reports					[-]
Report Name	**Status**	**Generated**	**Errors**	**Warnings**	**Infos**
Synthesis Report	Current	Tue Jun 2 20:09:12 2020	0	92 Warnings (77 new)	3 Infos (3 new)
Translation Report	Current	Tue Jun 2 20:10:04 2020	0	0	0
Map Report	Current	Tue Jun 2 20:10:27 2020	0	2 Warnings (2 new)	6 Infos (6 new)
Place and Route Report	Current	Tue Jun 2 20:10:41 2020	0	0	3 Infos (3 new)
Power Report					
Post-PAR Static Timing Report	Current	Tue Jun 2 20:10:49 2020	0	0	4 Infos (4 new)
Bitgen Report					

Secondary Reports		[-]
Report Name	**Status**	**Generated**
ISIM Simulator Log	Current	Tue Jun 2 20:09:56 2020

Date Generated: 06/01/2020 - 00:32:41

6.2 Synthesis Report Design Summary

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

---- Target Parameters
Output File Name                   : "parking_meter"
Output Format                      : NGC
Target Device                      : xc6slx16-3-csg324

---- Source Options
Top Module Name                    : parking_meter
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
```


```
---- 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

---------------------------------------------------
```

6.3 Map Summary

```
Design Summary
--------------
Number of errors:      0
Number of warnings:    2
Slice Logic Utilization:
  Number of Slice Registers:                    45 out of  18,224    1%
    Number used as Flip Flops:                  27
    Number used as Latches:                     18
    Number used as Latch-thrus:                  0
    Number used as AND/OR logics:                0
  Number of Slice LUTs:                        637 out of   9,112    6%
    Number used as logic:                      635 out of   9,112    6%
      Number using O6 output only:             474
      Number using O5 output only:               0
      Number using O5 and O6:                  161
      Number used as ROM:                        0
    Number used as Memory:                       0 out of   2,176    0%
    Number used exclusively as route-thrus:      2
      Number with same-slice register load:      0
      Number with same-slice carry load:         2
      Number with other load:                    0

Slice Logic Distribution:
  Number of occupied Slices:                   234 out of   2,278   10%
  Number of MUXCYs used:                       240 out of   4,556    5%
  Number of LUT Flip Flop pairs used:          638
    Number with an unused Flip Flop:           596 out of     638   93%
    Number with an unused LUT:                   1 out of     638    1%
    Number of fully used LUT-FF pairs:          41 out of     638    6%
    Number of unique control sets:               7
    Number of slice register sites lost
      to control set restrictions:              43 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:                        63 out of     232   27%

Specific Feature Utilization:
  Number of RAMB16BWERs:                         0 out of      32    0%
  Number of RAMB8BWERs:                          0 out of      64    0%
```


```
    Number of slice register sites lost
      to control set restrictions:              43 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:                        63 out of     232   27%

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:                4.31

Peak Memory Usage:  769 MB
Total REAL time to MAP completion:  18 secs
Total CPU time to MAP completion:   17 secs
```

Comment: The implementation requires a lot of flip-flops because they are important for the realization of SECONDS and counters assignment for each clock cycle.