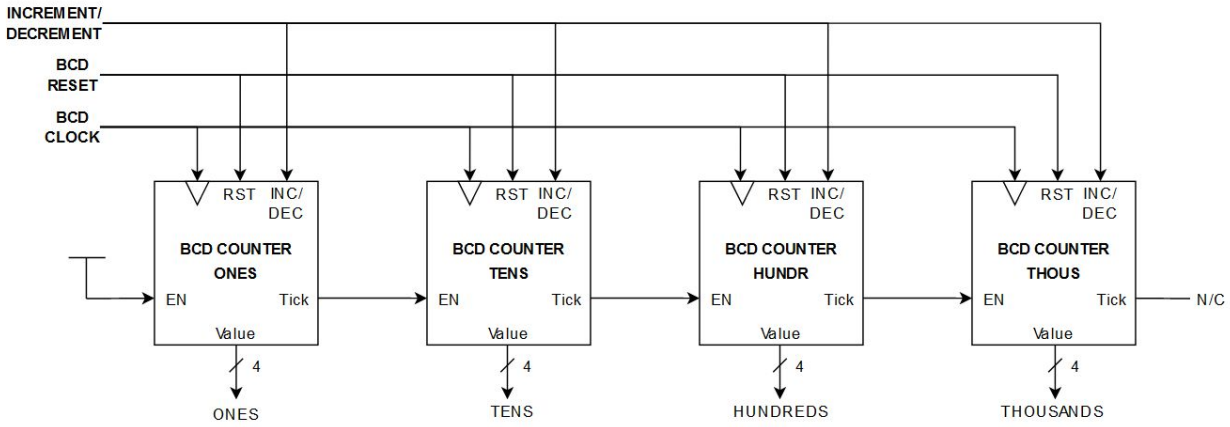
Lab 4 - Seven Segment Display Counter

Group A - Yuta Akiya, Kyle Le, Megan Luong
Prof. Aly
ECE 4304

Architecture

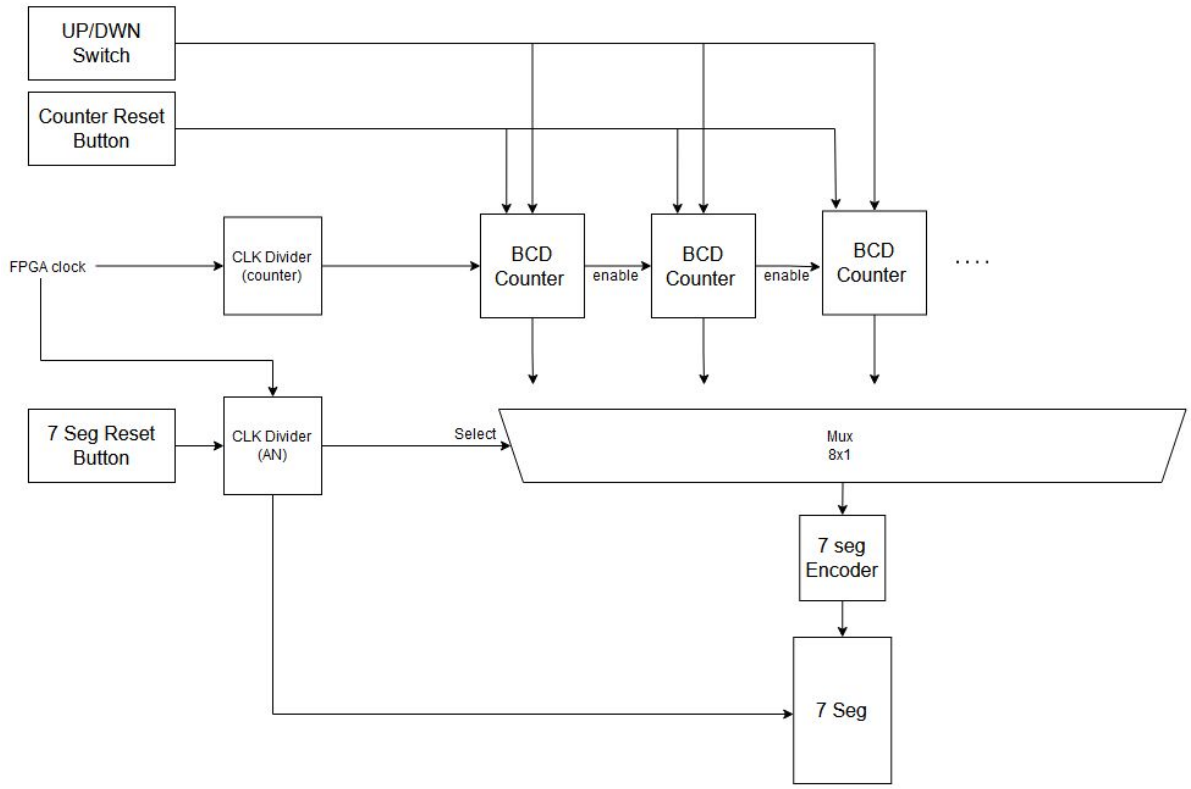
To implement a BCD counter for the 7-segment display, multiple one-digit BCD counters are cascaded together. In counting, the next significant digit changes when the previous one wraps around. For decrementing, this would be when going from 0 to 9. For incrementing, this would be when going from 9 to 0.

For the BCD Counter components, when this wrap around occurs, a tick value will go up for one BCD Clock cycle. This tick is fed into the enable of the next BCD counter, which will now increment or decrement. This could be cascaded for as many digits as desired. The following is the screenshot for a 4-digit BCD counter that ranges from 0-9999.



These 4-bit values are then placed into a mux, with the select being driven by the most significant bits of the 7-segment display slow clock. These bits also control which display is activated at a certain time. This combination allows the appropriate 4-digit BCD Counter value to be displayed on the appropriate display.

A separate clock divider circuit is used for changing the BCD counters and controlling the 7-segment displays. If the BCD counter changed value before the 7-segment display completely refreshed, then BCD values would not be properly displayed. To avoid this, two clock divider circuits were added to allow the 7-segment display enough time before the BCD counters change.



Code Details

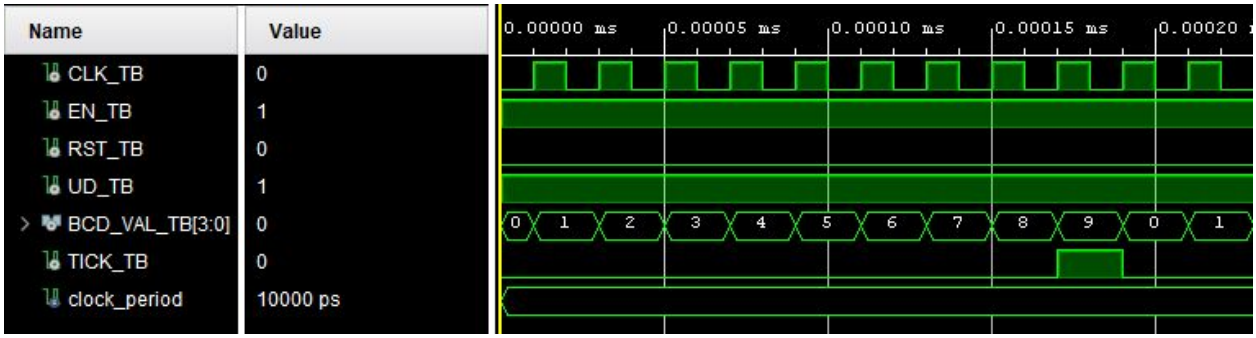
The cascaded BCD counters were created using a for-generate loop, with std_logic_vectors used to hold the enables and output values. The previous BCD counter would take from the current index, while outputting its values to the next index. This provided simple cascading without needing to re-write the code for each BCD counter needed. Upon reaching the maximum/minimum value while incrementing or decrementing, the counter rolls over to the minimum/maximum, displaying all zeros after the maximum value or all nines after the minimum value.

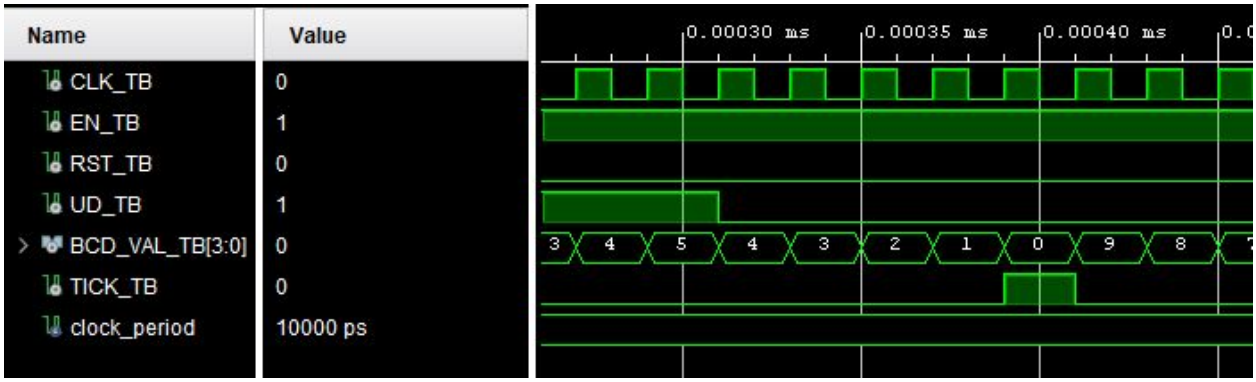
Corner Cases

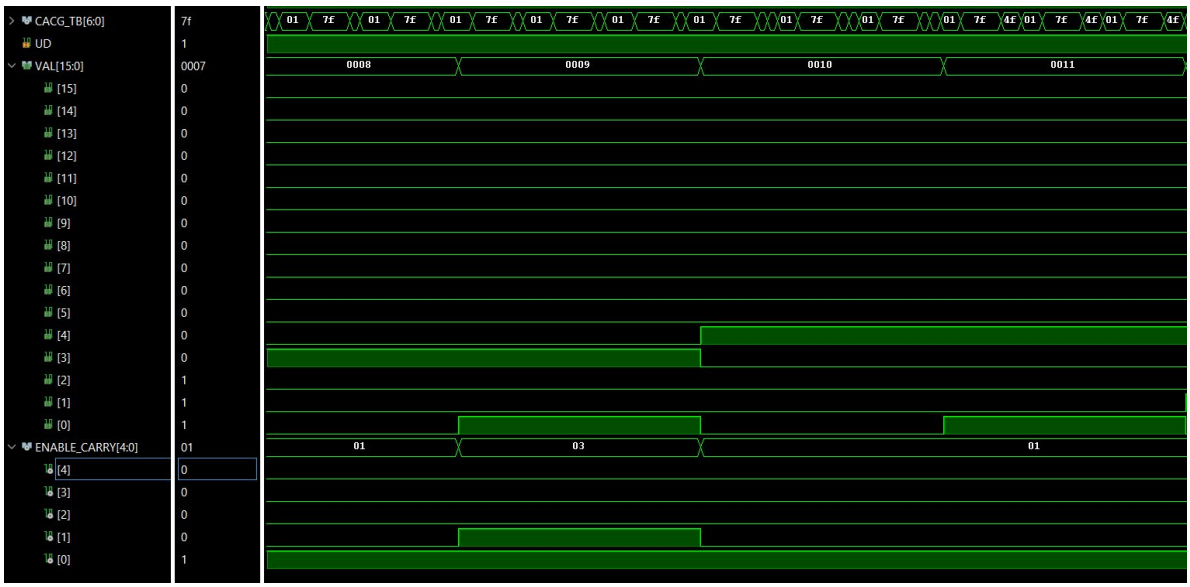
An important corner case for this project is when the seven segment display counter is moving the next digit up or down when a digit reaches a maximum or minimum value. For example, when the ones digit reaches 9 and the counter is on up mode, the tens digits needs to increment by one as the ones digit resets to show the value 10. The same thing applies when the counter is in down mode when a digit reaches 0. If the board is on value 10 and the counter is on down mode, the tens digit needs to decrement as the value 9 is loaded into the ones digit.

To deal with this corner case, each BCD counter has an output tick which is set to high when the counter reaches 9 while on up mode or 0 on down mode. This output tick is connected to the enable on the next counter (or the next digit), allowing the next counter to either increment or decrement depending on the counter's mode. With this design, the counter will be able to deal with this corner case no matter how many counters are implemented.

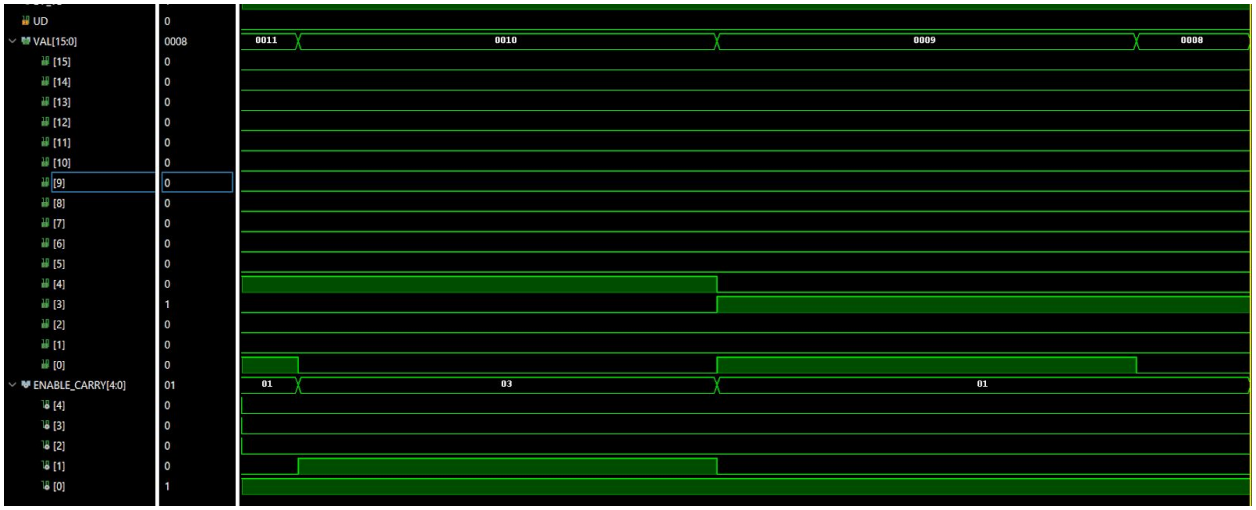
When the updown switch is flipped during the clock period, without the output tick, the counter may count below/above the minmax. Adding this output tick ensures that the counter does not increment to a value beyond the scope of BCD. When the BCD counters reach the maximum or minimum value, the counter "restarts" to the minimum, all zeros, or maximum, all nines, respectively.




Name
Value
CLK_TB
0
EN_TB
1
RST_TB
0
UD_TB
1
BCD_VAL_TB[3:0]
0
TICK_TB
0
clock_period
10000 ps
0.00030 ms
0.00035 ms
0.00040 ms
3
4
5
4
3
2
1
0
9
8



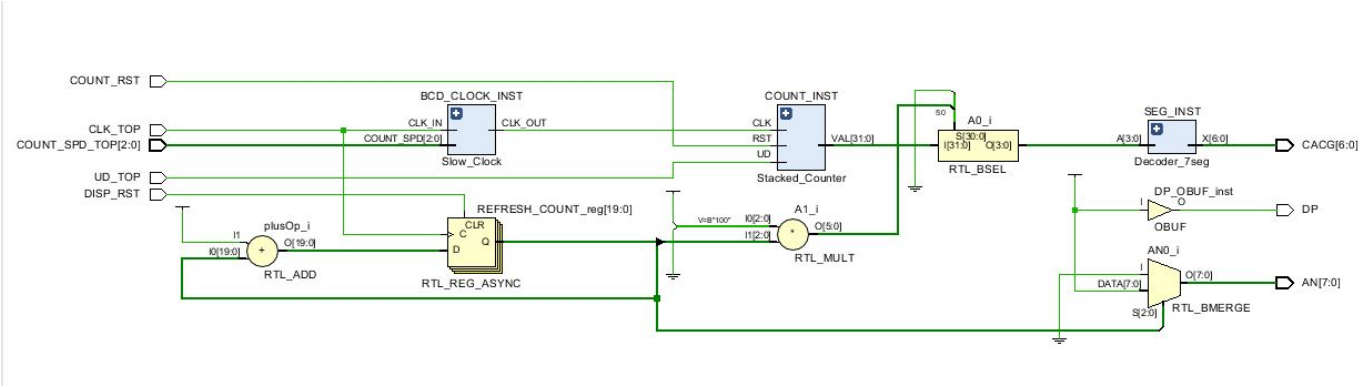
We can see that the first enable bit is always on for the ones digit counter but once the ones digit counter hits the value 9, the enable bit for the tens digit is set, the tens digit is incremented by one, and the value for the ones counter is reset to 0. Now, this cycle is repeated until both the ones and tens digit is 9. Then, the hundreds digit counter will start incrementing and so on. This cycle continues until the counter reaches its maximum value.



The same cycle applies when the counter is counting down (ud_tb = 0). When the counter value goes from 10 to 9, the enable bit for the tens digit is set, decrementing its value by one and the ones digit counter is set to 9.

Area/Resource Information

Elaborated Design of Entire System



Resource Usage of Entire System

Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	FF	BRAMs	URAM	DSP
✓ synth_1	constrs_1	synth_design Complete!								58	78	0.0	0	0
✓ impl_1	constrs_1	write_bitstream Complete!	5.306	0.000	0.237	0.000	0.000	0.114	0	58	78	0.0	0	0

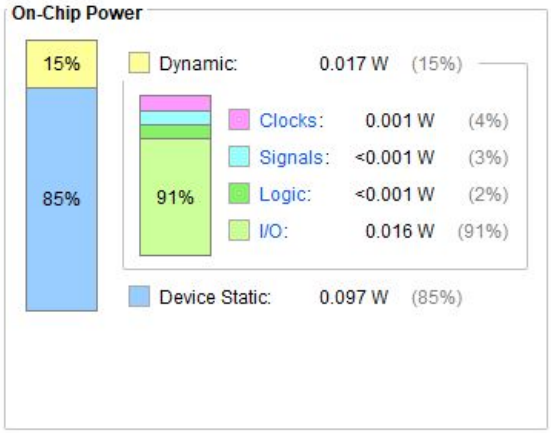
Power Usage Details

Summary

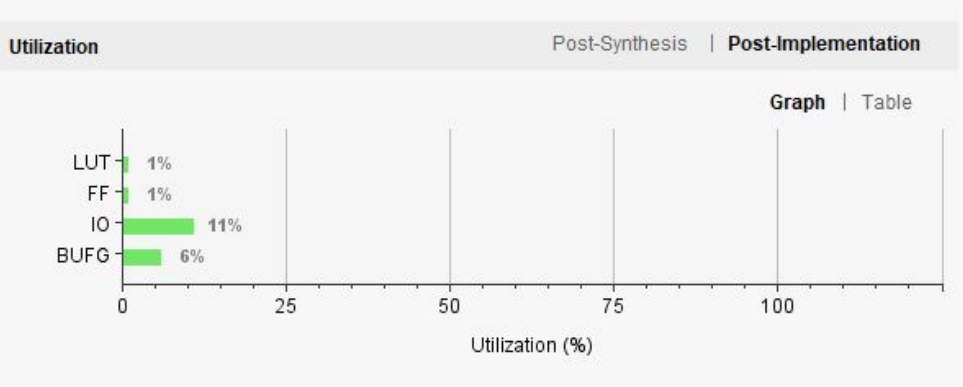
Power analysis from Implemented netlist. Activity derived from constraints files, simulation files or vectorless analysis.

Total On-Chip Power:	**0.114 W**
Design Power Budget:	**Not Specified**
Power Budget Margin:	**N/A**
Junction Temperature:	**25.5°C**
Thermal Margin:	59.5°C (12.9 W)
Effective ϑJA:	4.6°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Low

Launch Power Constraint Advisor to find and fix invalid switching activity



Utilization



Utilization Post-Synthesis | **Post-Implementation**

Graph | **Table**

Resource	Utilization	Available	Utilization %
LUT	58	63400	0.09
FF	78	126800	0.06
IO	23	210	10.95
BUFG	2	32	6.25