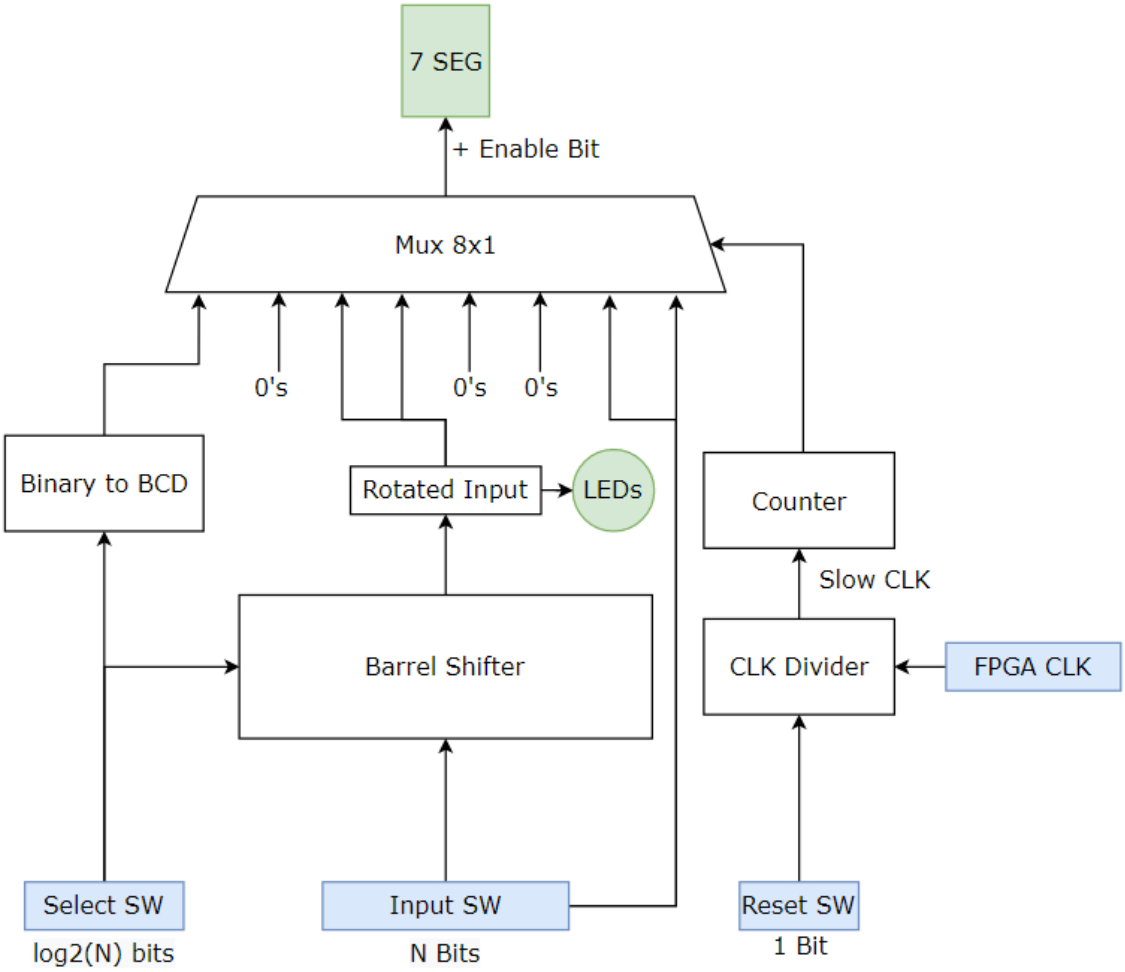
Lab 6 - Generic Barrel Shifter

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

Architecture:

The architecture of our lab 6 features two components: the barrel shifter component and the displaying on the 7 segs component. Since our past few labs have asked us to display things in BCD or HEX onto the 7 segment displays, a reusable and modular 7 segment displaying component was made. The displaying component includes the CLK divider which generates a slow clock for the counter to create a rotating select signal for the 8 to 1 mux. Each input into the mux is padded with an enable bit. For the inputs that will actually be displayed (select, inputs, outputs), the pad bit is 1. For fillers (permanent 0's), the pad bit is 0. Based on this pad bit, the 7 seg AN will be enabled or disabled for every mux output.

The barrel shifter will take in a generic sized input and rotate it X times based on the value of the select switches. The number of switches needed for the select is $\log_2(N)$ where N is the size of the input. The output will also be N bits but rotated X times. Since the select is the only input that is not BCD/HEX, it needs to be converted to BCD before being inputted into the display mux. For this demo where N=8, the barrel shifter output is displayed on 8 LEDs and two 7 segment displays, the input is displayed on two 7 segment displays, and the select (# of rotations) is displayed on one 7 segment display.



Within the barrel shifter module, a 2-D array is created to store the rotated values in. In a for loop, the module checks whether the ith select bit is '1', rotates the current array 2^i times (or 0 times if select[i] is 0) and stores it into the next layer of the 2-D array (i+1). The final output is stored in the last row of the 2-D array.

Ex. When SEL = 101 and N = 8

N

0	0	1	0	0	0	0	0	0	Original Input
1	1	0	0	0	0	0	0	0	Array[0] Rotated 2^0 * SEL[0] times
2	1	0	0	0	0	0	0	0	Array[1] Rotated 2^1 * SEL[1] times
SEL_SIZE	0	0	0	0	1	0	0	0	Array[2] Rotated 2^2 * SEL[2] times

Code Details:

Barrel Shifter

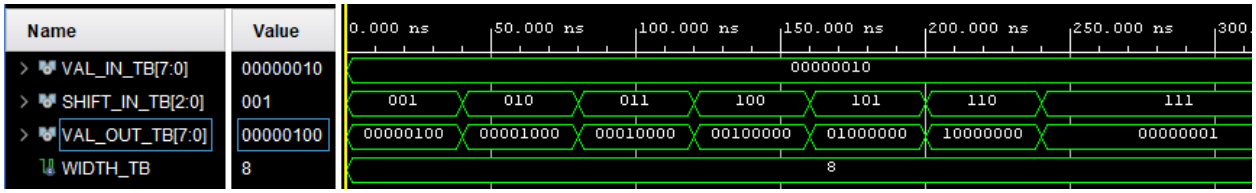
The design for this generic barrel shifter rotates a value, VAL_in, only to the left the amount of times inputted in SHIFT_IN. The stages of the barrel shifter, constant STAGES, is determined by taking the log2 of the size of the shift width, which is then used for the size of the 2D array. Using a for loop, for the number of stages, the value is shifted to the left while SHIFT_IN, the select bit, is on. The array holds the stages of the shift and the binary number at each stage.

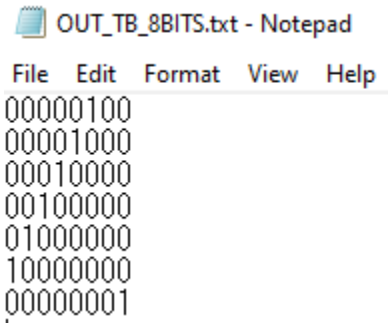
Corner Cases:

The only corner case that stands out in bit rotation is when the most significant bit gets rotated to the least significant bit. For example, when a 4 bit input 1000 rotates once to the left, it should become 0001. If we were to *shift* left (instead of rotating left), the output would be 0000. Verifying this corner case proves that we are rotating correctly and not shifting the bits. Other than this corner case, we can also test if the barrel shifter is completely generic by testing different input sizes.

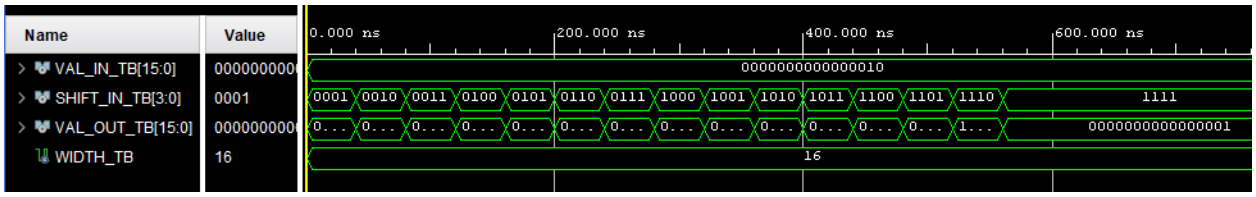
Using a text I/O test bench, we test this corner case with 8, 16, and 32 bit inputs.

8 Bits:

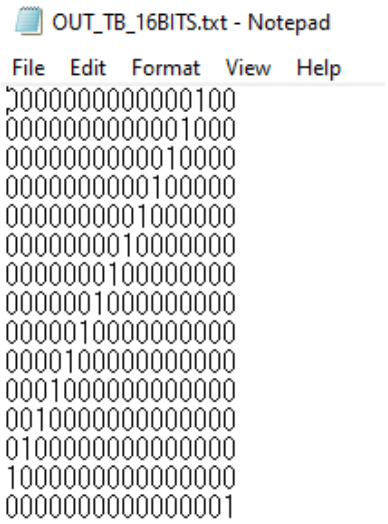




16 bits:

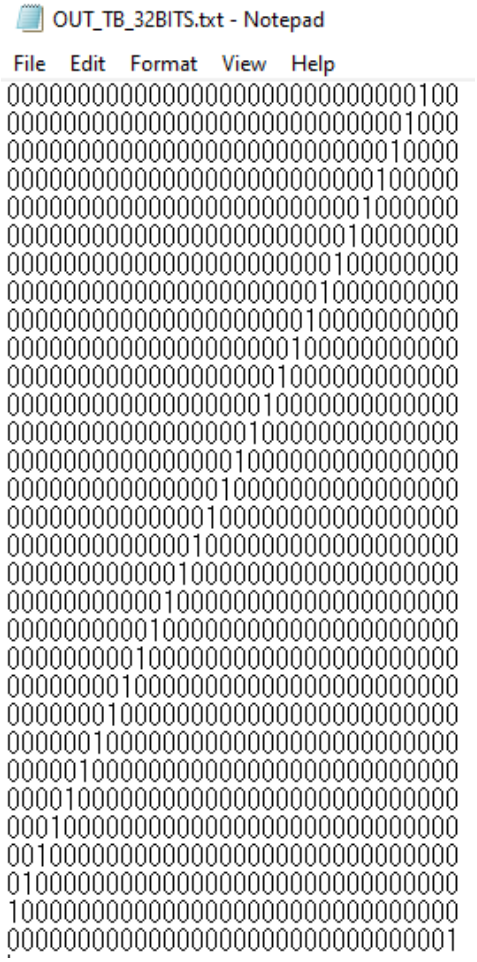


Hard to see in simulation so here is the output text file:



32 Bits:

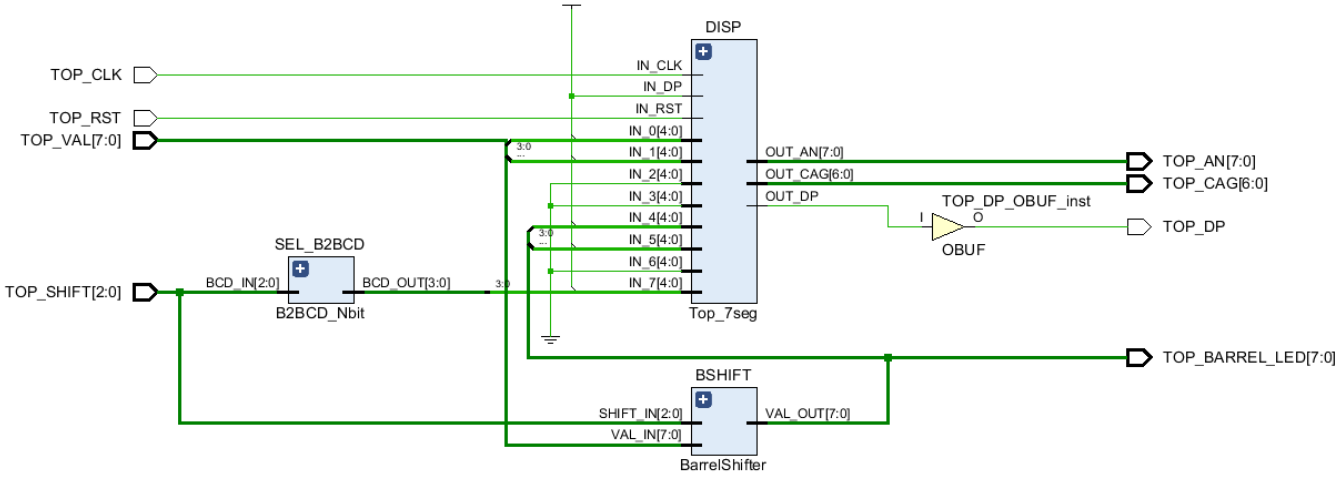
Simulation is impossible to read through a screenshot so here is the output text file:



In every test case (ever input size), we are able to see the 1 bit being rotated from the left most bit to the right most bit.

Area/Resource Information:

Elaborated Design of Entire System



Resource Usage of Entire System

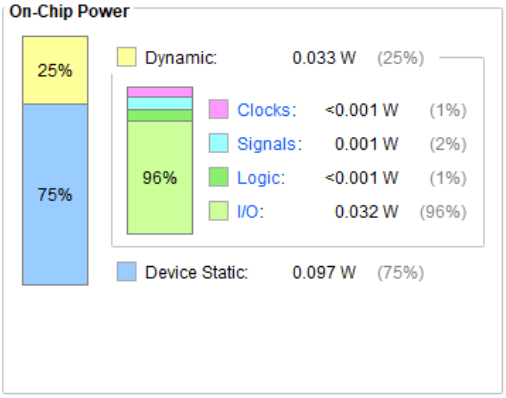
Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	FF	BRAMs	URAM	DSP
✓ synth_1	constrs_1	synth_design Complete!								49	21	0.0	0	0
✓ impl_1	constrs_1	write_bitstream Complete!	NA	NA	NA	NA	NA	0.130	0	49	21	0.0	0	0

Power Usage Details

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

Total On-Chip Power:	**0.13 W**
Design Power Budget:	**Not Specified**
Power Budget Margin:	**N/A**
Junction Temperature:	**25.6°C**
Thermal Margin:	59.4°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

Graph | **Table**

Resource	Utilization	Available	Utilization %
LUT	49	63400	0.08
FF	21	126800	0.02
IO	37	210	17.62
BUFG	1	32	3.13