IAS0600 LAB REPORT 4 CREEPING LINE

Kayode Hadilou ADJE

194360MAHM

Introduction

In some systems design, there is a need to store inputs story over need in order to perform specific operations. In that case, in addition to combinational logic, a memory element can also be used as components of the design. This lab was about more complex sequential logic circuit: a circuit with memory and combinational logic. At the end of the lab, it is expected to improve students' knowledge about designing sequential logic circuit.

Background

For this practical lab, the background sources I benefitted from are listed and detailed below:

-Lecture 8 and 9 Explanation and Slides: I relied on and used lecture 7 and 8 slides and explanations to understand more the notion of sequential systems and counters in particular;

-Lab Sheet: The explanations in the lab sheet was a guide for the completion of the tasks;

-The cumulated knowledge gained in previous lab work and first trainings practical classes about VHDL and Vivado were of great use.

-General understanding of seven segment that can be found in [1].

Workflow

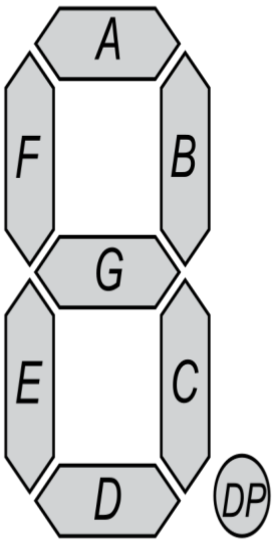
This lab was about implementing a creeping line running on eight seven segments LED displays. The lab was divided into 3 tasks:

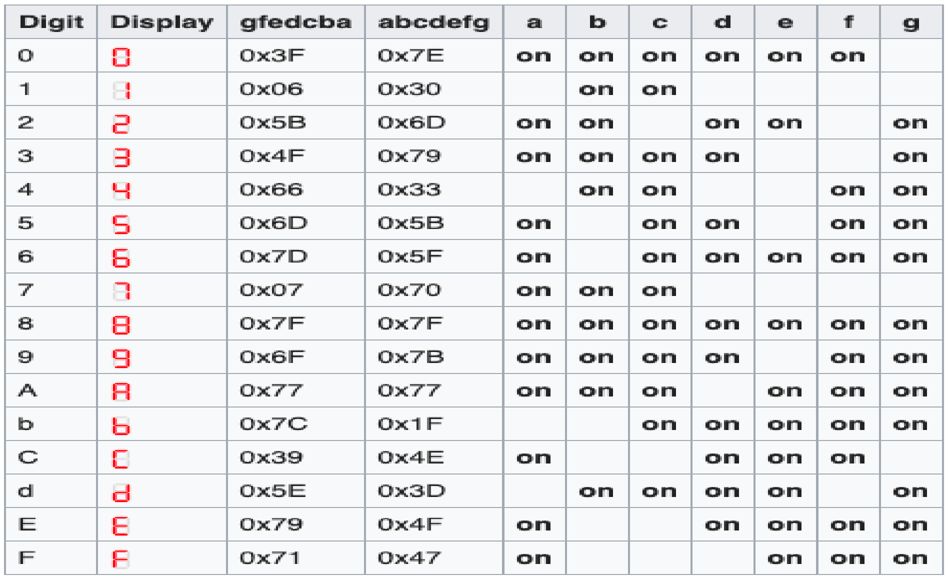
-implementation of a decoder able to converts binary value from 4 switches to a hexadecimal value shown on one seven-segment LED display.

-implementation of 4 seven segment LED displays each connected to a different set of 4 switches

-implementation of 8 seven segments LED displays displaying hexadecimal value from a 32-bit circular shift register initialized with a number.

Task1: Seven segment display is a form of electronic display device and another way of displaying number. Seven segment displays can be found in calculators, electronic meters and other electronics devices. In seven segment display, an hexadecimal number is represented using a combination of 7 bits (a to g). Below is the encoding for digits from 0 to F [1].





Digit	Display	gfedcba	abcdefg	a	b	c	d	e	f	g
0	0	0x3F	0x7E	on	on	on	on	on	on	
1	1	0x06	0x30		on	on				
2	2	0x5B	0x6D	on	on		on	on		on
3	3	0x4F	0x79	on	on	on	on			on
4	4	0x66	0x33		on	on			on	on
5	5	0x6D	0x5B	on		on	on		on	on
6	6	0x7D	0x5F	on		on	on	on	on	on
7	7	0x07	0x70	on	on	on				
8	8	0x7F	0x7F	on	on	on	on	on	on	on
9	9	0x6F	0x7B	on	on	on	on		on	on
A	A	0x77	0x77	on	on	on		on	on	on
b	b	0x7C	0x1F			on	on	on	on	on
C	C	0x39	0x4E	on			on	on	on	
d	d	0x5E	0x3D		on	on	on	on		on
E	E	0x79	0x4F	on			on	on	on	on
F	F	0x71	0x47	on				on	on	on

Figure 1: Seven Segment Encoding [1]

For example, to obtain the hexadecimal number A, bit D has to be set as off and the rest as on. The decoder built uses the same logic, it converts any number from 0 to 15 to its seven segment equivalent using *case ..when* statement in a process sensible to the inputs of the four switches. Since the inputs of seven segment LED displays are active low, *on* becomes *0* here and *off* becomes *1*. Anode inputs (active low as well) are used to select the seven segment LED display to use. In this case the third led was used so the 3rd anode bit was set to 0 and the rest to 1. Since the decoder is used in the next tasks, it was designed as a standalone source file and instantiated when needed. Below is the RTL schematic of this part of the lab work.

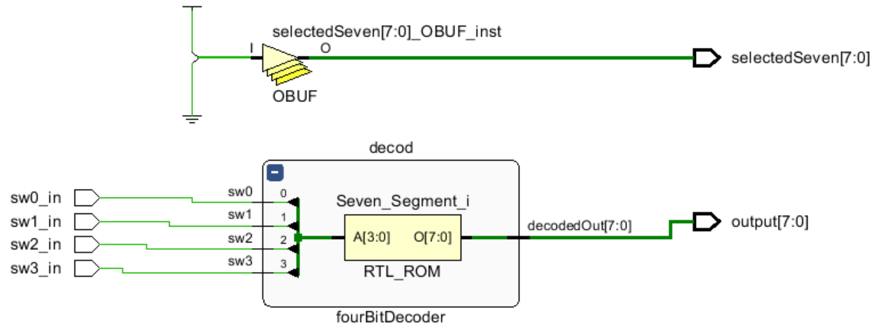


Figure 2: RTL schematic Task1

Task2: This task is about controlling four seven segments “simultaneously”. A multiplexer is used to allow a set of four bits coming from the switches to be passed to the decoder at 1ms clock time. Since this happen fast, every time input from switches change, the human brain interprets that as a simultaneous display. The board features an 100MHz clock that has been slow down 399999 times. The clock division is included in the counter description design and the output of the counter is used as enabled bit for the 15x1 multiplexer. The output of the multiplexer is decoded to hexadecimal seven segment format same as in task 1. Note that the output of the counter is also used to determine which anode to display to at the specific time. For that, another decoder (2 bits) is used. The corresponding RTL schematic resumes what has been said.

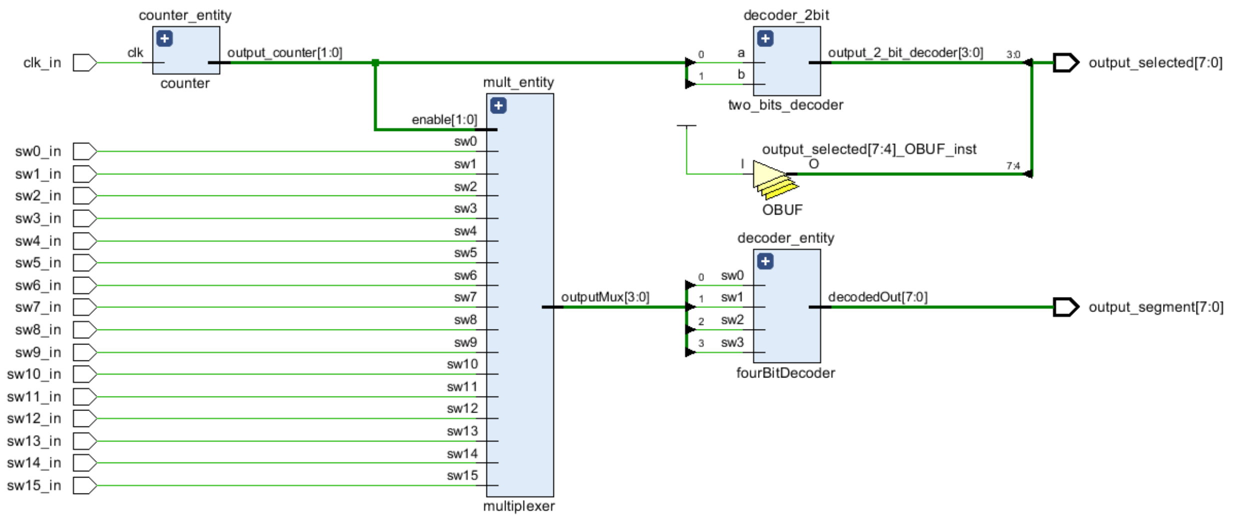


Figure 3 Task 2 RTL schematic

Task 3: In task 3, it is required to use a 32-bit input circularly shifted by 4 at specific time. A 32-bit binary numbers corresponding to (01234567) hex has been used and is circularly shifted by 4 every half a second (500). Meaning clock system was slow down 4999999 times. So, every half a second the shift register changes positions of its bits. The multiplexer runs with a clock of 1 MS and detect these changes very fast. The rest of the mechanism is similar to task2. Below is the RTL schematic of the task.

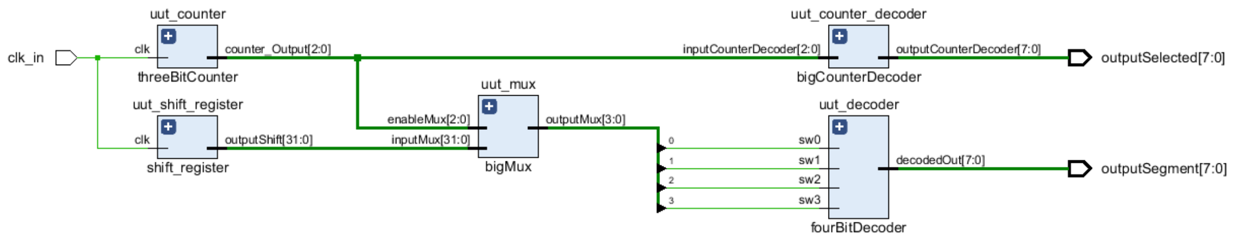


Figure 4: RTL Task 3

Results and Discussion

In task 1, a screenshot added below is the waveform result of the testbench simulation. To verify the correctness of the description, at each point of change of switches values, I checked if the *s_output* value is the seven-segment representation of the binary number obtained from the combination of switches. The results were as expected.

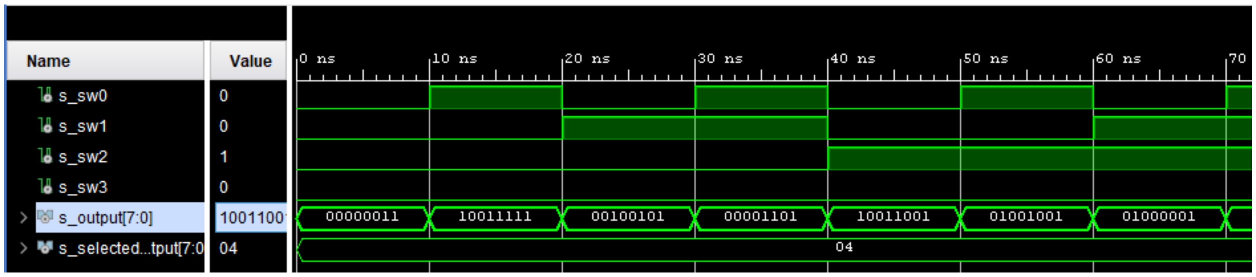


Figure 5: Waveform Task 1

In task 2, I checked if the set of switches corresponding to *s_output_counter* is the one represented in seven-segment form by the signal *output_segment.* The screenshot below confirms the correctness of the description.

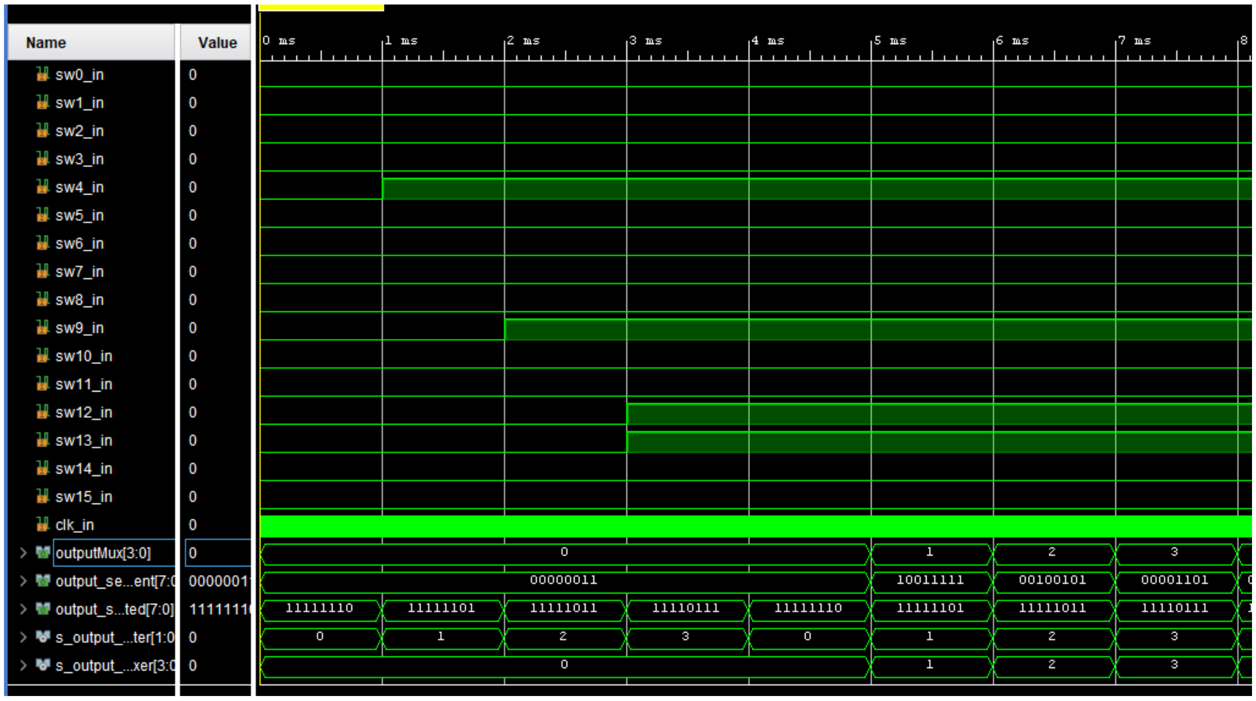


Figure 6: Waveform Task 2

In task3, the description is expected to perform a 4-bit circular right shift every 500MS. Also the output of the seven segment display is expected to match with the corresponding set of 4 bits in the 32 initial number. The result is shown in below figure and represents the waveform of testbench 3.

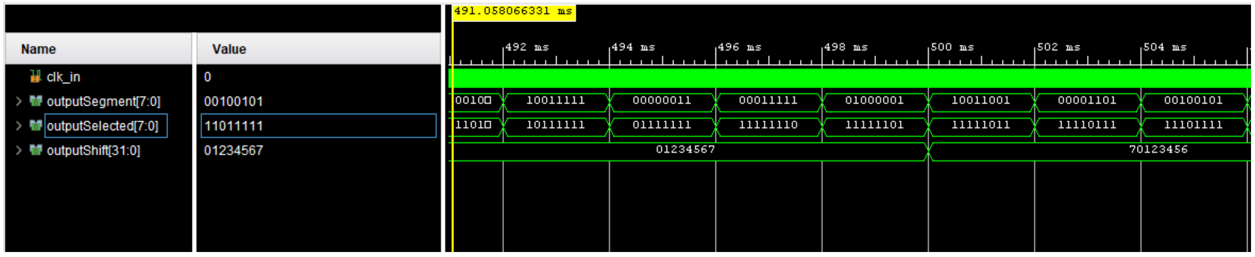


Figure 7: Waveform Task 3

Conclusion

In this lab, I have improved my understanding of sequential design logic and used my knowledge to describe a little more complex system than before using sequential circuits. The lab was completed successfully.