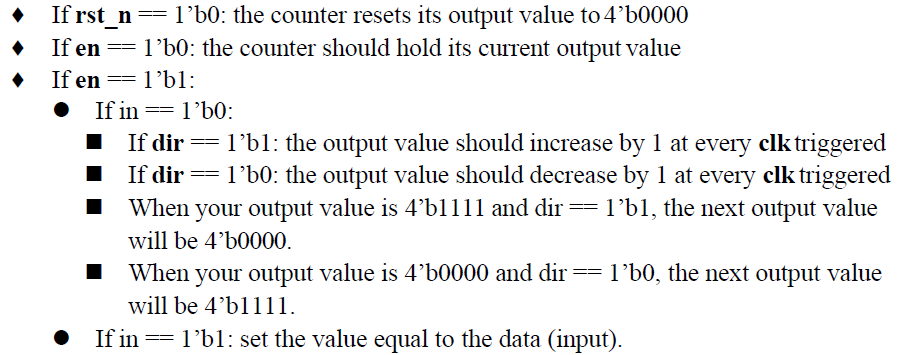
|  |
| --- |
| **EECS 2070 02 Digital Design Labs 2019**  **Lab 3** |
| **學號：107062361 姓名：許珉濠** |

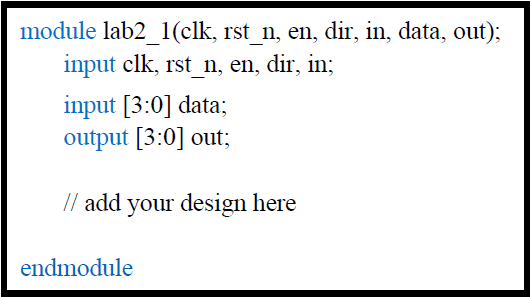
1. 實作過程

* lab2\_1

The code is running based on given requirement.



The code is designed based on given template.

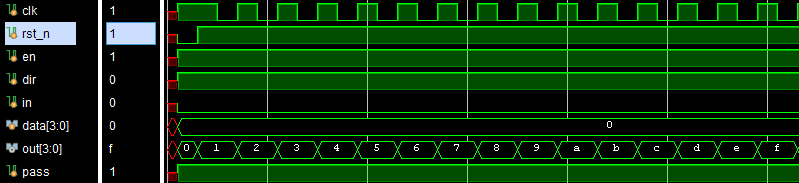


I designed it with behavioral modeling method with always block that will only run in positive-edge-triggered clock or rst\_n value changes.

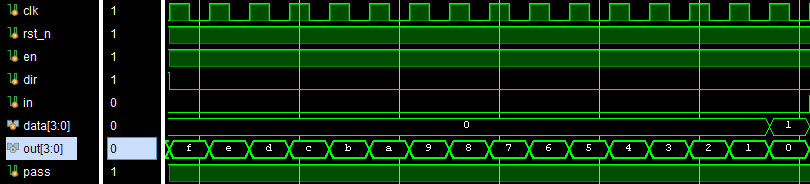
* lab2\_1\_t

The testbench is designed similar to previous lab 1 before. The testbench is filled with all possible testcase in 4-bit binary counter which counts onwards from 0 to 15 or backwards from 15 to 0. In addition, there are special case when the counter is counted onwards and backwards. When the current counter number is 15, the counter number will become 0 after incremented by one. On the other hand, if the current counter number is 0, the counter number will become 15 after decremented by one.

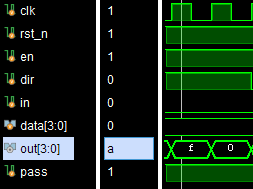
The counter counts onwards from range 0 to 15, if **en** == 1, **in** == 0, **dir** == 1.



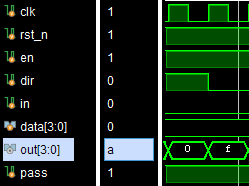
The counter counts backwards from range 15 to 0, if **en** == 1, **in** == 0, **dir** == 0.



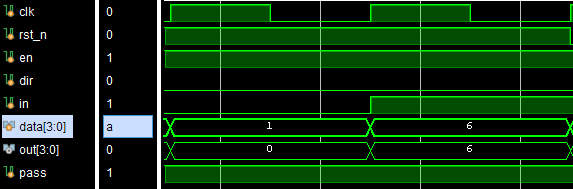
The counter number become 0 when the previous number 15 is incremented by one.



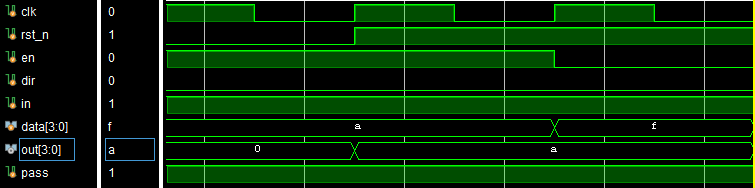
The counter number become 15 when the previous number 0 is decremented by one.



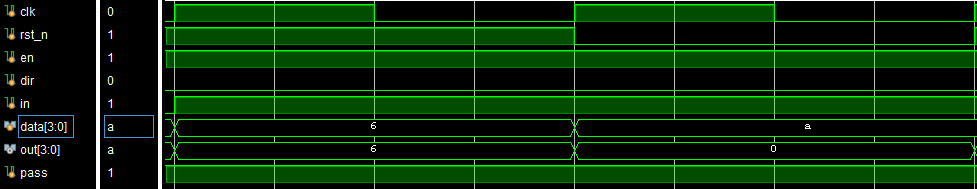
In this code, it is possible to replace the 4-bit **output** with the given 4-bit **data** when **clk** is positive-edge-trigged, **en** == 1, and **in** == 1.



The output value replacement won’t work when the **en** == 0.



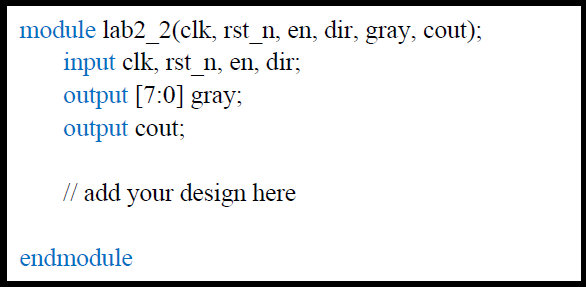
When **rst\_n** == 0, it will reset the **output** become 0.



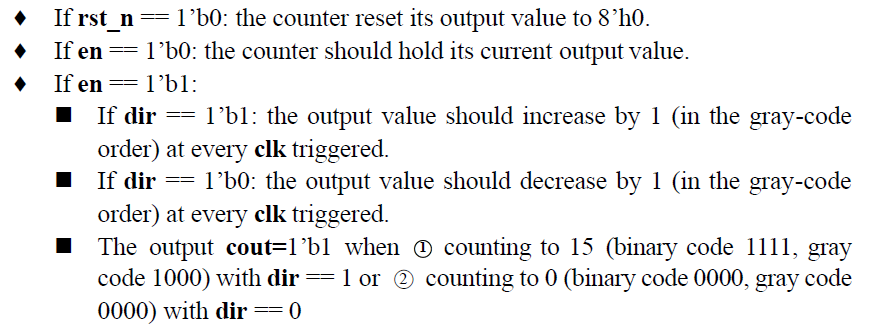
.

* lab2\_2

Given template



The code is designed in behavioral modeling method with negative-edge-triggered and reset in always block. The 1-digit gray-code counter is designed as the base of 2-digit gray-code counter. The counter runs according to below requirement.

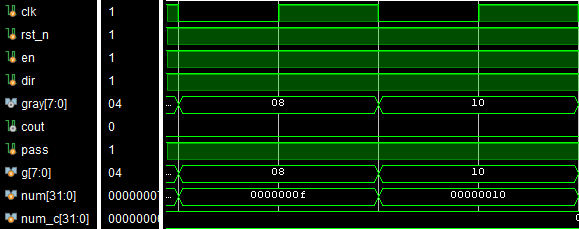


First, the counter is designed as 4-bit binary counter which is similar to lab2\_1. Subsequently, the 4-bit binary counter output is converted into 4-bit gray-code counter through formula

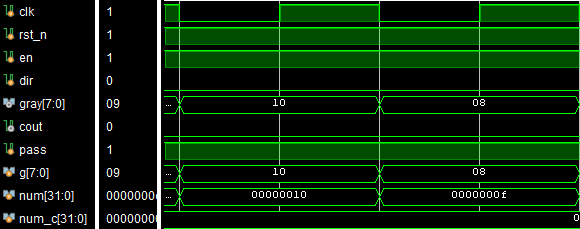
“**gray[3:0] = {gray0[3], gray0[3]^gray0[2], gray0[2]^gray0[1], gray0[1]^gray0[0]};**“

with gray0 as 4-bit binary counter. The formula will be applied to both two grey-code counter. The second 4-bit gray-code counter is incremented by one when the first 4-bit gray-code counter number is 15 and incremented by one become 0. It is vice versa in decrement condition which means the second 4-bit gray-code counter is decremented by one when the first 4-bit gray-code counter is decremented from 0 become 15.

Second 4-bit gray-code counter increment by one, when first 4-bit gray-code counter is incremented by one if current number is 15.

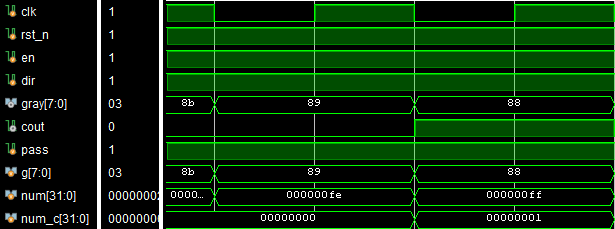


Second 4-bit gray-code counter decrement by one, when first 4-bit gray-code counter is decremented by one if current number is 0.

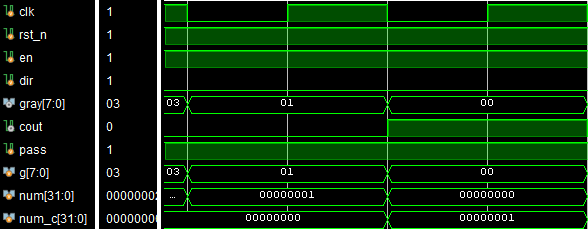


Since this is 8-bit gray-code counter, the **cout** will become 1 if only satisfies two conditions. When the counter counts onwards from 254 to 255 with **dir** == 1, or counts backwards from 1 to 0 with **dir** == 0.

Counter count onwards from 254 to 255, **cout** = 1.

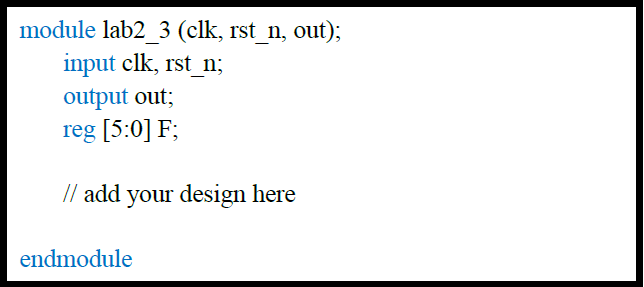


Counter count backwards from 1 to 0, **cout** = 1.

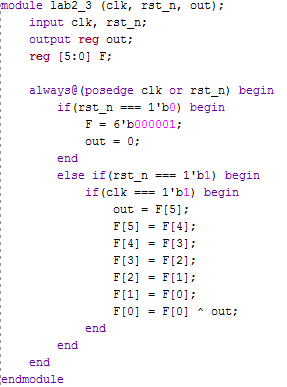


* lab2\_3

With the given template, the code is designed in behavioral modeling style with positive-edge-triggered clock.



The picture is the LFSR designed code based on given logic diagram.



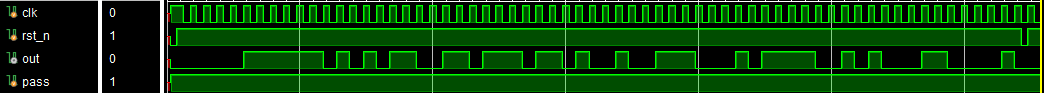
Logic diagram



There are requirement that if **rst\_n** == 0, then **F** will be reset into **6’b000001**. If the clock is positive-edge-triggered, it will shift the register flip-flop F to the right. The **F[0]** will get the value from **XOR** gate between previous **F[0]** and **out**.

* lab2\_3\_t

This testbench is designed similar to lab1’s testbench. It is designed with 63 different possible input combination which picture is shown below.



1. 學到的東西與遇到的困難

I felt scared when Professor said that this lab is asking us to create a testbench. Because the testbench that I have seen before is really complicated. I though that this homework will be very tough. After I went home, I started doing it. In the beginning, my mind went blank because I didn’t have any idea to start it. Then, I decided to do the lab2\_1 since it is easier. When I finished writing the lab2\_1, I took a look at lab2\_1\_tb again. I contemplate the way for long time and browse any testbench. Eventually, I decided to use lab1 testbench as the sample testbench. I write with the similar coding style in the lab2\_1 testbench. After writing a short and simple testcase, I am surprised that it is working. I keep going to think and write about all the possible testcase. Thus, the testbench is done although it is not really a good way. Hopefully, I can write better code in the next lab.

1. 想對老師或助教說的話

It is nicer if there is some example of how to create testbench.