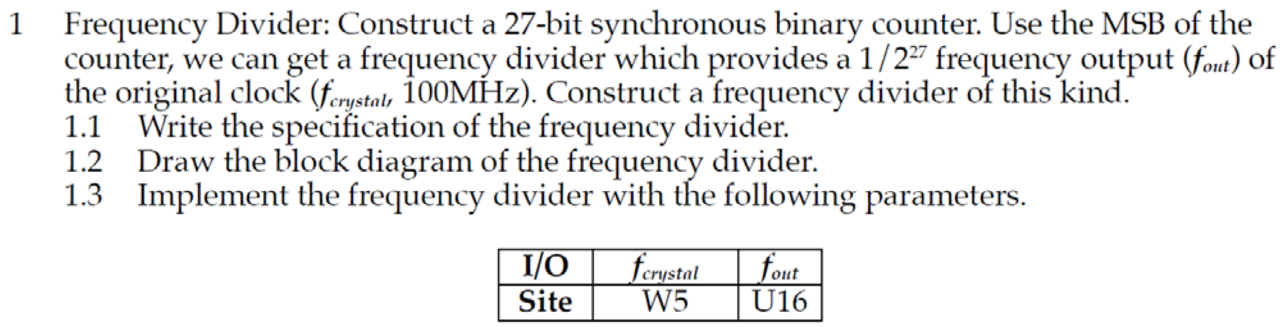


1 Frequency Divider: Construct a 27-bit synchronous binary counter. Use the MSB of the counter, we can get a frequency divider which provides a $1/2^{27}$ frequency output (f_{out}) of the original clock ($f_{crystal}$, 100MHz). Construct a frequency divider of this kind.

1.1 Write the specification of the frequency divider.

1.2 Draw the block diagram of the frequency divider.

1.3 Implement the frequency divider with the following parameters.

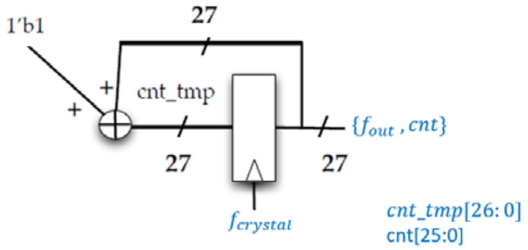
I/O	$f_{crystal}$	f_{out}
Site	W5	U16

Design Specification:

Input: f_crystal, rst_n

Output: f_out

Block diagram:



Design Implementation:

I/O pin assignment:

f_out—U16

f_crystal—W5

rst_n—R2

Discussion:

In this problem, we are required to get approximatedly 1Hz signal from the FPGA's original 100MHz signal and show the output on a LED. So I designed a 27-bits binary counter and used its MSB, whose frequency is 1/2^27 times the original clock frequency from FPGA board. So, f_out is about 100M/2^27 = 0.74Hz. Hence, our human eyes are able to observe the blinking LED representing f_out instead of LED being always on.

2 Frequency Divider: Use a count-for-50M counter and some glue logics to construct a 1 Hz clock frequency. Construct a frequency divider of this kind.

2.1 Write the specification of the frequency divider.

2.2 Draw the block diagram of the frequency divider.

2.3 Implement the frequency divider with the following parameters.

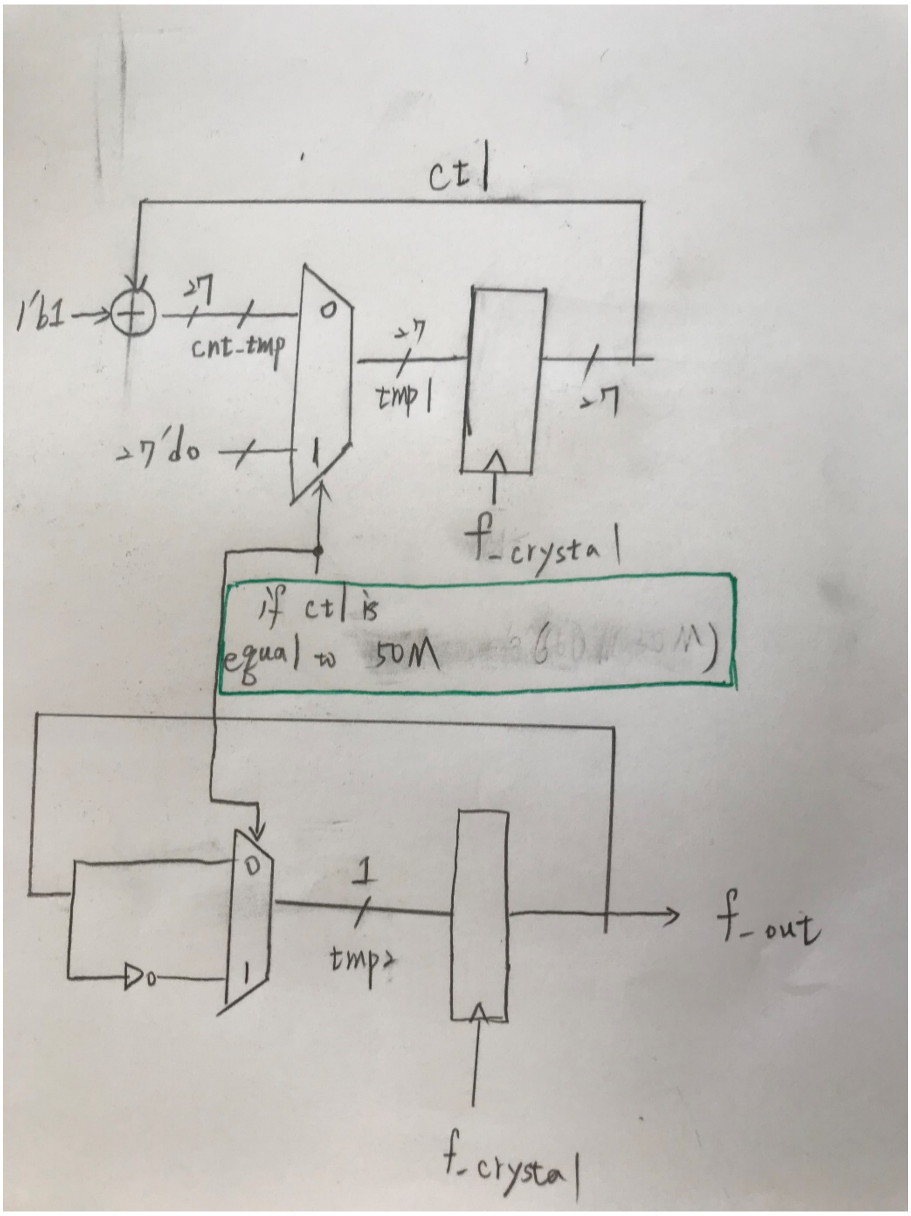
I/O	$f_{crystal}$	f_{out}
Site	W5	U16

Design Specification:

Input: f_crystal, rst_n

Output: f_out

Block diagram:



Design Implementation:

I/O pin assignment:

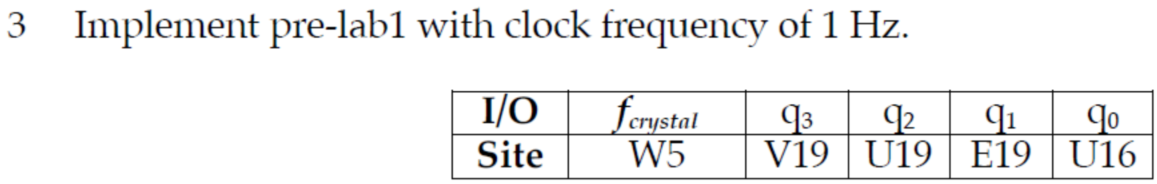
f_out—U16

f_crystal—W5

rst_n—R2

Discussion:

In this problem, we are required to get **exactly** 1Hz from FPGA board using frequency divider architecture. As clock frequency from FPGA is 100MHz, I designed a counter, which counts from 0 to 50M. When counter counts to 50M, f_out will be assigned as ~f_out (1->0 or 0->1), otherwise, f_out is unchanged. Therefore, I need a MUX to choose f_out value as shown in the block diagram. I also need MUX to select counter value. When it reaches 50M, I need to reset the value to 0, otherwise, it's keep being added by 1'b1.



3 Implement pre-lab1 with clock frequency of 1 Hz.

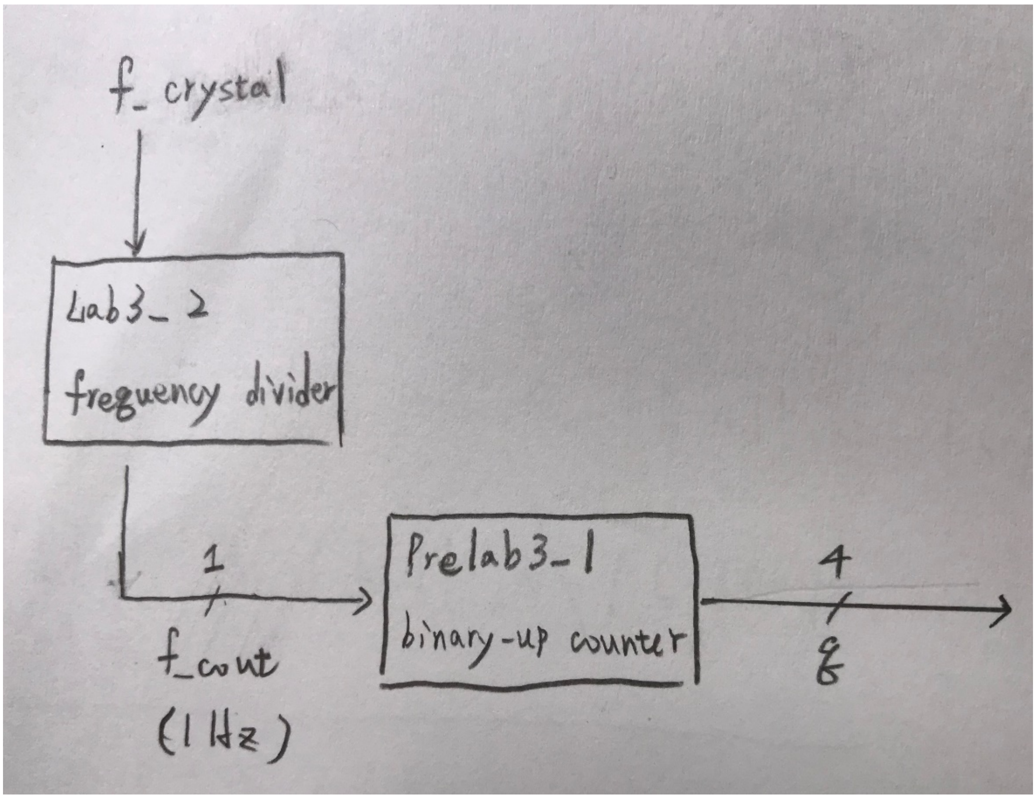
I/O	$f_{crystal}$	q_3	q_2	q_1	q_0
Site	W5	V19	U19	E19	U16

Design Specification:

Input: f_crystal, rst_n

Output: q[3:0]

Block diagram:



Design Implementation:

I/O pin assignment:

f_crystal—W5

rst_n—R2

q[3]—V19

q[2]—U19

q[1]—E19

q[0]—U16

Detail of frequency divider and binary-up counter has been illustrated at report of lab3_2 and prelab3_1

I used a top module to include module from lab3_2 and prelab3_1 and used wire to connect two modules, such as f_out in the block diagram.

Discussion:

In this problem, we used the 1Hz frequency derived from lab3_2 to display the 4-bit binary-up-counter result from 0000 to 1111 on LED. We need 1Hz clock to control the 4-bit binary counter. As our eyes are only able to observe the different patterns of LED when they're changing in 1Hz frequency. If the frequency is too high, what we could see is 4 always-on LEDs.

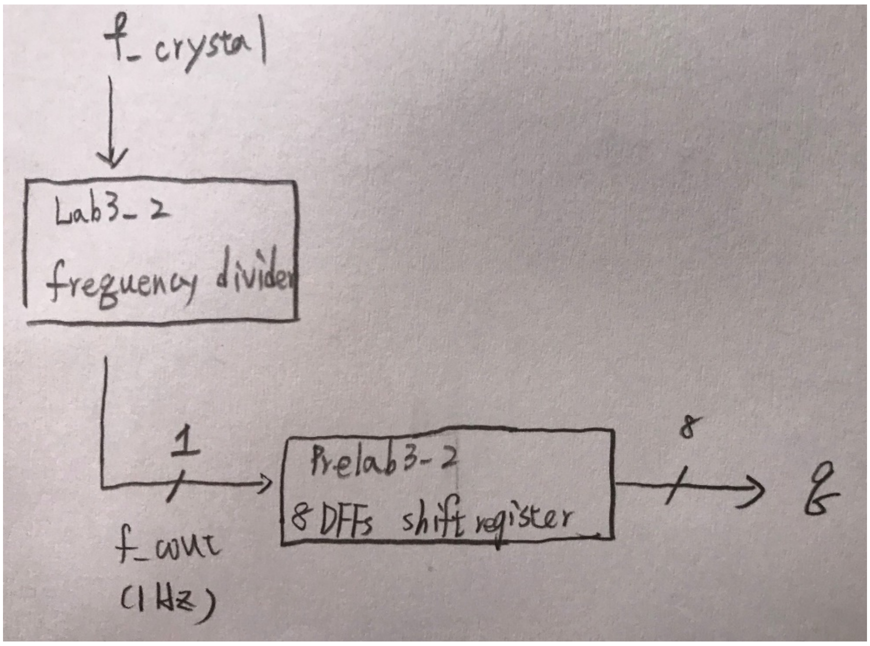
4. Implement pre-lab2 with clock frequency of 1 Hz. The I/O pins can be assigned by yourself.

Design Specification:

Input: f_crystal, rst_n

Output: q[7:0]

Block diagram:



Design Implementation:

I/O pin assignment:

f_crystal—W5

rst_n—R2

V14—q[7]

U14—q[6]

U15—q[5]

W18—q[4]

V19—q[3]

U19—q[2]

E19—q[1]

U16—q[0]

Detail of frequency divider and shift register has been illustrated in the report of lab3_2 and prelab3_2.

Similar to lab3_2, I also used a top module to include module from lab3_2 (frequency divider) and prelab3_2 (shift register). Also, I connect them using wire as well, such as f_out in the block diagram.

Discussion:

In this problem, we need to implement prelab3_2 and show the result on the LED on the FPGA board. To do so, we need to use frequency divider to get 1Hz clock frequency to control shift register so that our eyes are able to observe the different patterns of LED. As output (q[7:0]) are initialized as 8’b0101_0101 after reset, there are only two patterns of LED, 01010101 & 10101010.

Conclusion for Lab3:

In this lab, I have learned the concept of counter and how it is being used in the frequency divider, which is very important so that our eyes are able to observe the change of LED or 7-segment display pattern. Also, it’s my first time to handle Flip Flop, which is a indispensable element in sequential circuit. After the lab3, I am more familiar with how to describe behavior of Flip Flop in Verilog. And Finally, I have learned that I can combine small modules to form a complicated module. By doing so, I could only take care of each module and make the connection between different modules are well-defined, which is easier compared to writing **a single** module with complicated functions.

Reference:

Handout given by professor, from which I have learned the importance of counter and how to describe Flip Flop in Verilog.