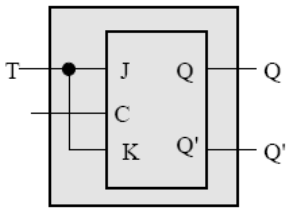
Laboratory 7 Design a Frequency Divider and Programming a FPGA

Background:

A frequency divider can be constructed from T flip-flops and AND gates where T-type flip-flop is shown in Figure 1. A T flip-flop is obtained from a JK flip-flop by tying the J and K inputs together to form the T input. The input of each T flip-flops is set to 1 to produce a toggle at each cycle of the clock input.



T	Q(t+1)
0	Q
1	Q'

Function table

Figure 1

For each two toggles of the first cell, a toggle is produced in the second cell, so its output is at half the frequency of the first. The output of the third cell is 1/8 the clock frequency. The same device is useful as a counter.

Figure 2 shows schematic design for four-bit frequency divider by using four T-type flip-flops.

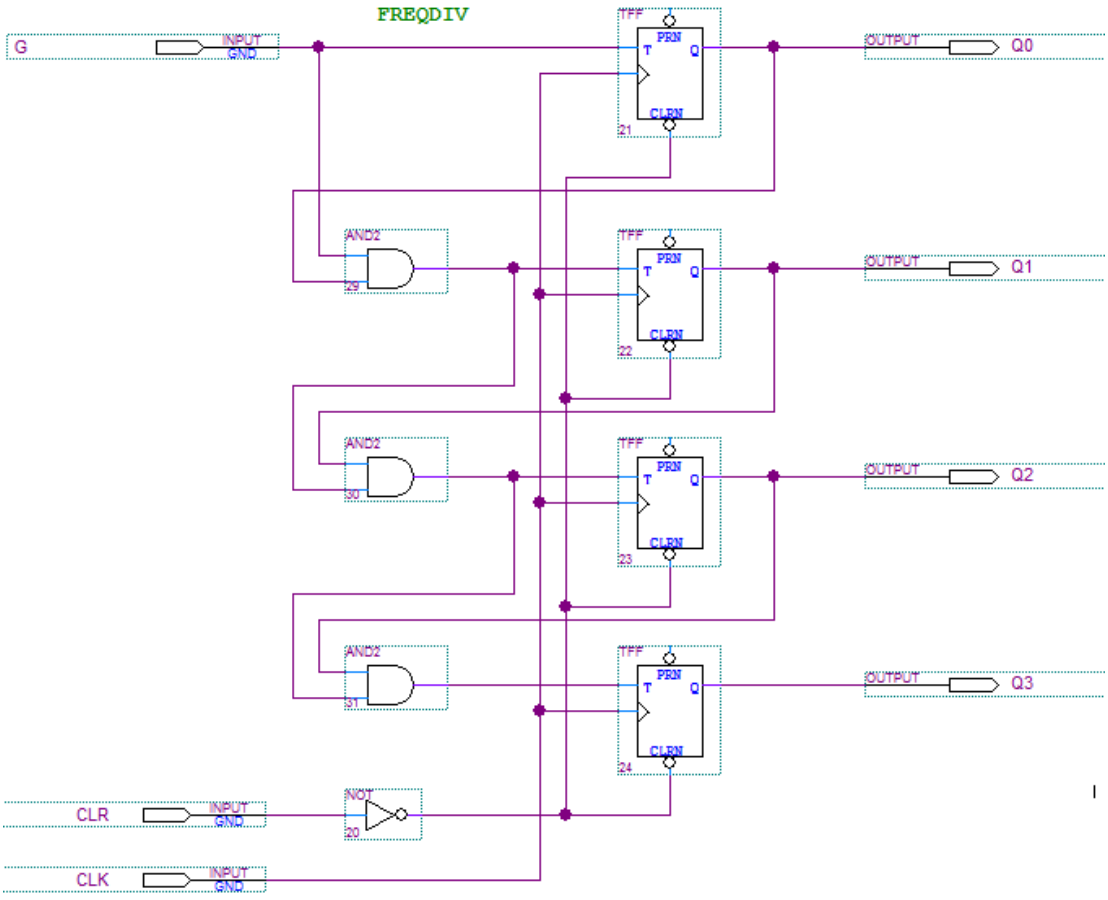


Figure 2

Figure 3 shows *FourBitFreqDiv.vhdl* to use the following library in your VHDL design entry

```
1   library ieee ;
2   use ieee.std_logic_1164.all ;
3   use ieee.std_logic_arith.all ;
4
5   entity FourBitFreqDiv is
6       port (
7               CLK : in std_logic ;
8               G: in std_logic;
9               CLR: in std_logic;
10              Q0: out std_logic;
11              Q1: out std_logic;
12              Q2: out std_logic;
13              Q3: out std_logic);
14      end FourBitFreqDiv ;
15
16  architecture arc of FourBitFreqDiv is
17      signal count:  unsigned (3 downto 0);
18  begin
19      process (CLK, CLR, G)
20      begin
21          if (CLR = '1') then
22              count <= "0000" ;
23          -- positive triggering
24          elsif (CLK' event and CLK = '1') then
25              if (G = '1') then
26                  count <= count +1;
27              end if;
28          end if;
29      end process;
30
31      Q0 <= count(0);
32      Q1 <= count(1);
33      Q2 <= count(2);
34      Q3 <= count(3);
35
36  end arc;
```

Figure 3

Figure 4 is the simulation result. Check the relation among clock, Q0, Q1, Q2, and Q3

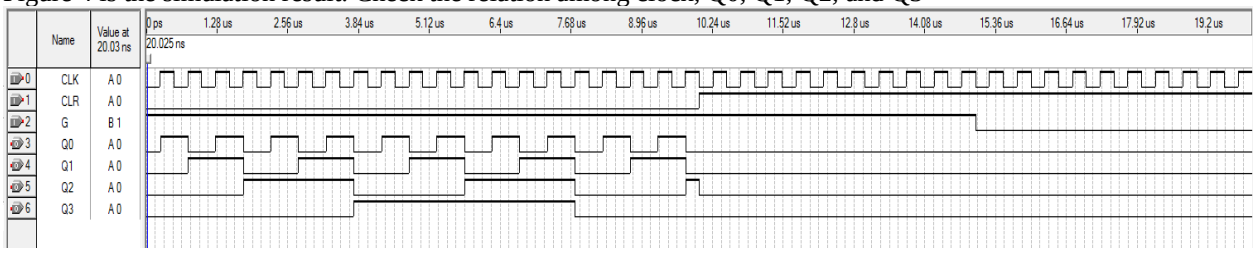


Figure 4

Suppose a clock frequency is clock = 24kHz (you can set different frequency by using function generator)
The frequency of the 2^0 output line (Q0) is 1/2 of the input clock line (12kHz)
The frequency of the 2^1 output line (Q1) is 1/4 of the input clock line (6kHz)
The frequency of the 2^2 output line (Q2) is 1/8 of the input clock line (3kHz)
The frequency of the 2^2 output line (Q3) is 1/8 of the input clock line (1.5kHz)

Project:

Consider the extension of the circuit. Create an 8-bit s counter which VHDL approach. The counter **decrements** its count on each *positive edge* of the clock if the *Enable* signal is asserted. The counter is reset to 0 by using the *Clear* signal.

8-bit frequency divider

1. Write a *VHDL* file that defines an **8-bit counter (8-bit frequency divider)** by using the structure depicted in Figure 3 (vhdl) or LPM freqdiv (Figure 5), and compile the circuit. What is the frequency relation between the clock and eight outputs from an eight bit frequency divider, respectively?
2. Create a waveform vector and simulate your system to verify its correctness of the design of 8-bit frequency divider.
3. Augment your VHDL file to use the GPIO as the *Clock* input (the input from function generator), switches SW_1 and SW_0 as *Enable* and *Clear* inputs, and 8 pins from GPIO to display the divided frequencies in oscilloscope.
4. All output signals from GPIO must be connected to your breadboard in order to be observed from oscilloscope or displayed by external LEDs
5. Make the necessary pin assignments and compile the circuit.
6. Programming FPGA and test your implementation. All input and output frequencies should be measured from oscilloscope. Develop a table and record the observable frequencies. Take photos from your experiments.

	Name	Component	Pin Location		Name	External Component	GPIO pin	Component	LED Pin Location
Input		SW0	PIN_N25	Output	Q4	Frequency observed on oscilloscope	GPIO 4	LEDG4	?
Input	..	..		Output	Q2	Frequency observed on oscilloscope	GPIO 2	LEDG2	?
Input	CLK	Function generator		Output	Q0		GPIO 0	LEDG0	PIN_AE22

7. You may use the following library in your VHDL design entry:

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_unsigned.all;
…
```

8. Quartus II software provide a library. It is called Library of Parameterized Modules and can be found in the /others. Right-click in a new block design (bdf) workspace and choose insert> symbol **freqdiv**

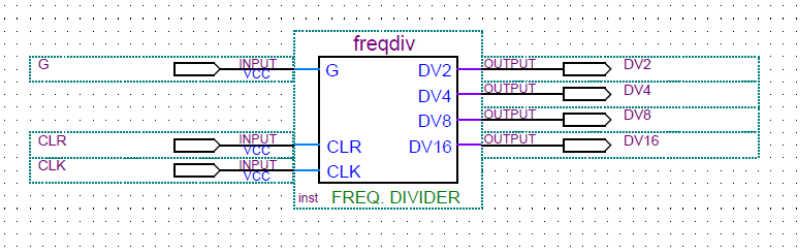


Figure 5

9. Design a frequency counter. Input clock will be 50MHz (internal clock), output signal will have at least two output frequencies: one is in Hz level (1Hz~ 10Hz) and one is in KHz level. These two frequency signals can be observed by oscillator respectively. (You cannot observe 50MHz signal; however, you can observe signals in Hz and KHz). These frequency signal (Hz and KHz) will be used in your future projects.

For your report:

- The problem written in English
- The flowchart or function table to solve the problem if it is necessary
- The design entry included (VHDL or Schematic)
- The simulation result for designed digital component
- The analysis for the simulation
- The pin assignment- the table for assigning the circuit inputs and outputs to specific pins on the FPGA
- The test table you designed to record and verify the designed circuit on hardware
- Picture taken for your test if it is necessary
- The conclusion