

# Introduction and Requirement

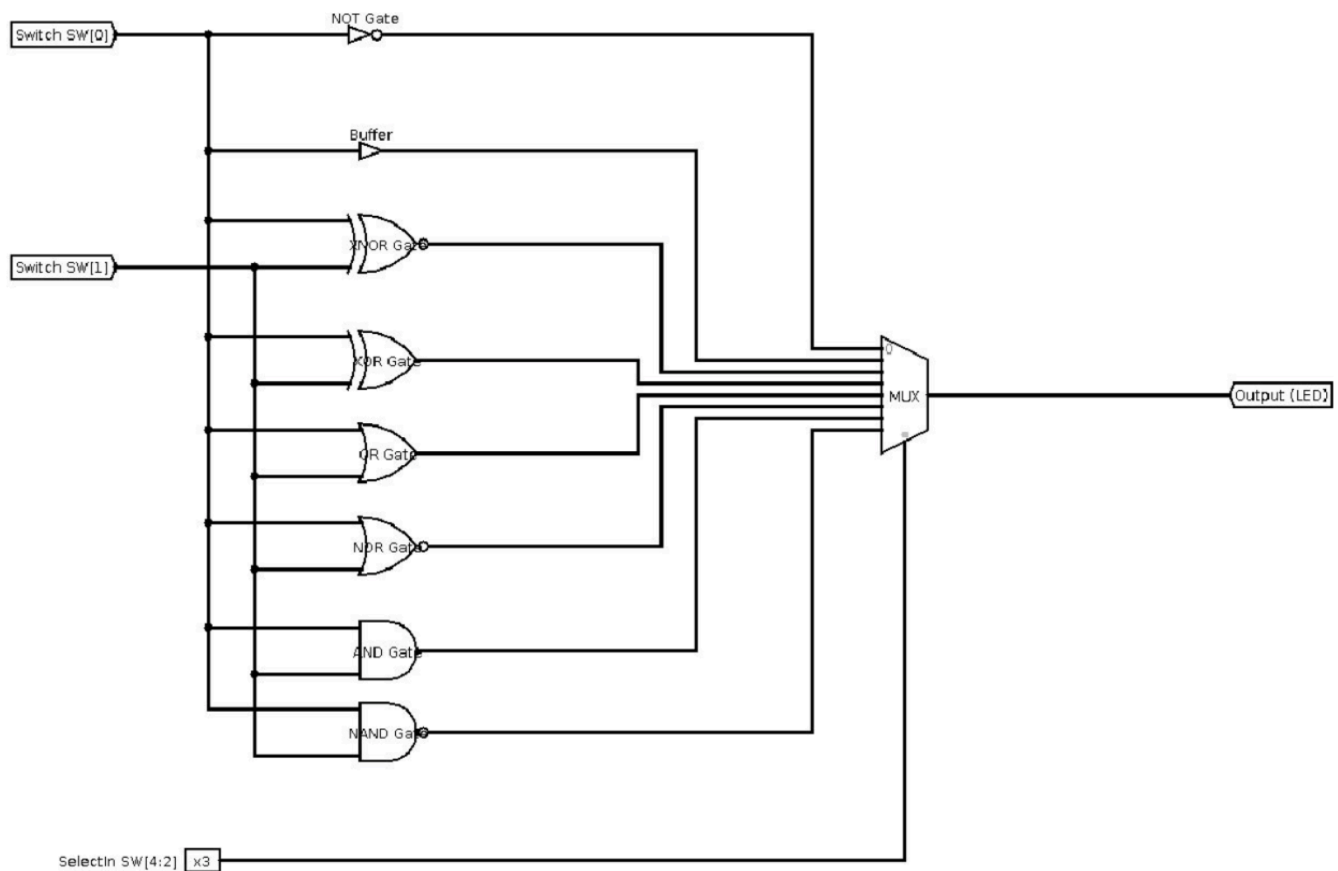
## Background Information

This project is aim to provide basic practice to help students to learn Verilog which is HDL language, and get familiar with using Xilinx ISE to simulate and synthesize.

There are mainly two parts for this project.

## 1. FPGA design with Combination Circuit

The first part is Combination Circuit Example. It requires to design the hardware and the testbench for a combinational logic circuit to be implemented on an FPGA board. The schematics of the combinational logic is as below:



It consists of eight gates, NAND, AND, NOR, OR, XNOR, XOR, NOT, Non-inverting buffer (BUFF) gates and a multiplexer at the output to select between the outputs of the gates. Basically, this part is about FPGA design workflow with a combinational logic design.

In this part, students need to provide their own test bench for the simulation and accurately describe the combinational logic circuit and the procedures to the simulation. The accepted testbench output should be from Xilinx Isim simulator.

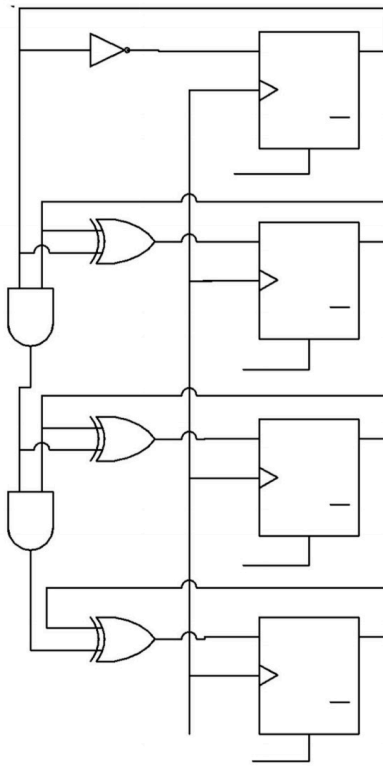
## 2. Sequential Circuit

---

The second part is Sequential Circuit. It requires to design and perform simulations, and explain the real implementations. There are mainly three steps for this part, of several small projects revolving around counters. Students are expected to provide the code as your design, perform the simulations, and implement a small design revolving around counters.

1. Design a 4-bit counter using gate-level hardware description based on schematics, and perform the simulation.

The below figure shows the schematics of a 4-bit counter with D flip-flops, students are required to translate the schematics into Verilog code. Each gate should be mapped to an operator, and each flip-flop should be mapped to one line in the edge-sensitive always block. Remember to use **non-blocking** assignment `<=` in edge-sensitive always blocks.



2. Design the same counter using a higher-level abstraction, and perform simulations.

```
// Example Verilog code for the counter

reg [3:0] a;

always @ (posedge clk)

    if (rst)

        a <= 4'b0000;

    else

        a <= a + 1'b1;
```

Test the provided code in simulation to check if it gives the same outputs as previous.

3. Create a simple clock divider using a counter and simulate this clock and describe our findings.

Create an flashing LED with a frequency of 1-Hz. Since we are not working with a real FPGA at the moment, you should create an output wire and consider “1” as its light-on state, and “0” for the light-off state. In your test bench, design a 10kHz clock and use that.

# Design Description

## Part1

---

For the design of architecture of the part 1, I first setted up the input and output, then assigned 8 wire data types to generate the eight outputs for the eight logic gates. And then generate them as the inputs for the 8:1 multiplexer. So this part is just about partice of using basic verliog.

## Part2

---

### 4-bit counter:

This part does not need to design the arcitecure of the module. Basically I translated the schematics into the code. I setted up the input and output, and use always blocks and non-blocking statement to generate the 4 output for the four D-flip-flops.

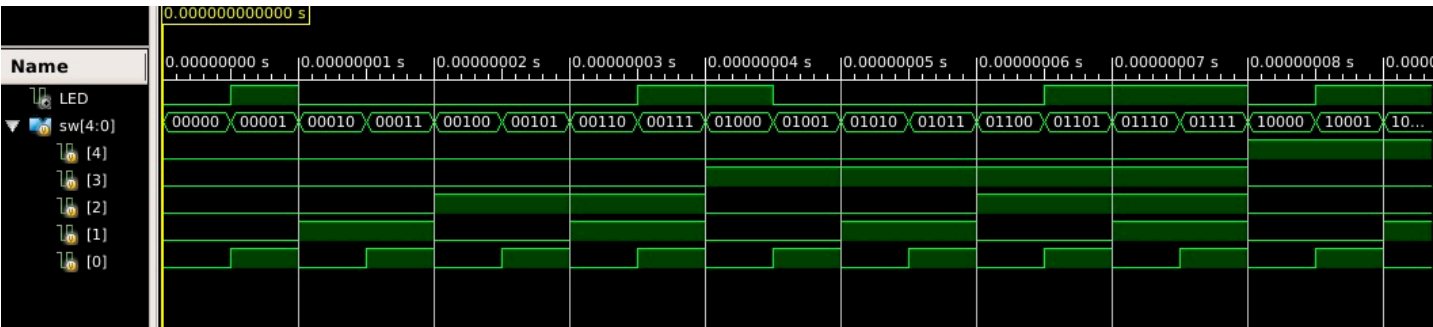
### Clock Divider:

The structure I desgin for this clock divider is as below. Convert 10k Hz to 1 Hz. So it should be 10000 clock cycles before `clk_1khz` goes to '1' and returns to '0'. In other words, it takes 5000 clock cycles for `clk_1hz` to flip its value. The basic idea is that, when the counter arrive at the 5000 clock cycles, the counter resets and flip the `clk_1khz` signal.

# Simulation Documentation

## Part1 Simulation

To assess the correctness of the implementation of part1, I created my own testbench to validate the design. The below figure is the simulation waveforms.



Test Case	Input	Output
1	SW = [00000]	LED = 0
2	SW = [00001]	LED = 1
3	SW = [01011]	LED = 0
4	SW = [00011]	LED = 0
5	SW = [00100]	LED = 0
6	SW = [10101]	LED = 0
7	SW = [00111]	LED = 1
8	SW = [01000]	LED = 1
9	SW = [11111]	LED = 0

## Part 2

The below figure is the simulation waveforms for my translated design code.

The below figure is the simulation waveforms for the given design code.

It gives the same output as the previous one!

Test case	Input clk	Input rst	Output count[3:0]	Output a[3:0]
1	1	0	0010	0010
2	0	1	0000	0000
3	1	1	0000	0000

For the part of the clock divider, the project requires us to create 10khz Clock input. For this, i convert 10khz to half a cycle which should be 50000 ns ( $1s = 1e9ns$ ,  $1e+9 / 10k = 100000$ ,  $100000/2 = 50000ns$ ). The I put `always #50000 clk_in=~clk_in;` in the testbrench to create the input.

The below figure is the simulation waveform for the clock docker.

# Conclusion

## Design Summary

---

This project is not hard, but it takes you time to learn the ISE if you are new. You probably would waste time in how to use this software. And one thing i learned is that before writing the code, making drafts for the architecture of the circuit would be really healful. I spend a day for the clock divider, it is because I didn't draft anything, I just think in my head, which wasted the time.

## Difficults encountered and solution:

---

At the beginning of the project, since I am not familiar with Xilinx ISE, there are some issues of license which takes me a couple of hours to solve. When I am doing simulation, this error message showed up in the console window.

Console

```
NFO: There is another simulation running in the same directory. Using database file name isim1.wdb.  
Sim P.20131013 (signature 0xfbc00daa)
```

```
-----  
NFO:Security:61 - The XILINXD_LICENSE_FILE environment variable is not set.  
NFO:Security:63 - The LM_LICENSE_FILE environment variable is not set.  
NFO:Security:68 - For more information or for assistance in obtaining  
a license, please run the Xilinx License Configuration Manager  
          (xlcm or "Manage Xilinx Licenses").  
NFO:Security:68a - user is ise, on host localhost.localdomain.  
NFO:Security:15 - A feature for ISIM was found but is for the wrong hostid.  
ERROR:Security:14 - No feature was available for 'ISIM'.
```

```
invalid host.
```

```
The hostid of this system does not match the hostid  
specified in the license file.
```

```
Feature:      ISIM
```

```
Hostid:       08002768c935
```

```
License path:
```

```
home/ise/.Xilinx/Xilinx.lic/opt/Xilinx/14.7/ISE_DS/ISE//data/*.lic/opt/Xilinx/14.7/ISE_DS/ISE//coregen/core_licenses/Xilinx.lic/opt/Xilinx/14.7/ISE_DS/ISE//  
egen/core_licenses/XilinxFree.lic/opt/Xilinx/14.7/ISE_DS/EDK/data/core_licenses/Xilinx.lic:
```

```
FLEXnet Licensing error:-9,57
```

```
For further information, refer to the FLEXnet Licensing documentation,  
available at "www.flexerasoftware.com".
```

```
-----  
WARNING: A full ISim License cannot be checked out due to the issues listed above. Please use Xilinx License Configuration Manager to fix these issues  
in order to check out a full ISim license.
```

```
WARNING: ISim will run in Lite mode. Please refer to the ISim documentation for more information on the differences between the Lite and the Full  
version.
```

```
This is a Lite version of ISim.
```

```
Time resolution is 1 ps
```

```
Sim>
```

To solve this, I acquire my new license of Xilinx, get the .lic file put it into the share folder, then load them in Xilinx. The problem solved.

Second, an issue I met when I was translating the 4-bit counter. Warning reports showed as below:

```
1  WARNING:Xst:2972 - "/home/ise/152A/Project1/Part2_4bit_counter.v" line 36. All outputs of
   instance <D1> of block <D_flipflop> are unconnected in block <Part2_4bit_counter>. Underlying
   logic will be removed.
2  WARNING:Xst:2972 - "/home/ise/152A/Project1/Part2_4bit_counter.v" line 37. All outputs of
   instance <D2> of block <D_flipflop> are unconnected in block <Part2_4bit_counter>. Underlying
   logic will be removed.
3  WARNING:Xst:2972 - "/home/ise/152A/Project1/Part2_4bit_counter.v" line 38. All outputs of
   instance <D3> of block <D_flipflop> are unconnected in block <Part2_4bit_counter>. Underlying
   logic will be removed.
4  WARNING:Xst:2972 - "/home/ise/152A/Project1/Part2_4bit_counter.v" line 39. All outputs of
   instance <D4> of block <D_flipflop> are unconnected in block <Part2_4bit_counter>. Underlying
   logic will be removed.
```

## Question answer

---

.ucf files are User Constraint Files.

UCF lists all the available pin mappings in the FPGA in the following format:

```
Net "your_signal_name" LOC = XX | IOSTANDARD = LVCMOS33; # More details about the pin
```

I would use it when I need to assign FPGA pins to the switches and leds so they will be connected to the correct ports on the board.

## Design Summary Report

---



<b>Part1 Project Status (04/20/2020 - 06:23:55)</b>			
<b>Project File:</b>	Project1.xise	<b>Parser Errors:</b>	No Errors
<b>Module Name:</b>	Part1	<b>Implementation State:</b>	Placed and Routed
<b>Target Device:</b>	xc6slx16-3csg324	<b>Errors:</b>	No Errors
<b>Product Version:</b>	ISE 14.7	<b>Warnings:</b>	No Warnings
<b>Design Goal:</b>	Balanced	<b>Routing Results:</b>	All Signals Completely Routed
<b>Design Strategy:</b>	Xilinx Default (unlocked)	<b>Timing Constraints:</b>	
<b>Environment:</b>	System Settings	<b>Final Timing Score:</b>	0 (Timing Report)

<b>Device Utilization Summary</b>	<b>[-]</b>			
<b>Slice Logic Utilization</b>	<b>Used</b>	<b>Available</b>	<b>Utilization</b>	<b>Note(s)</b>
Number of Slice Registers	0	18,224	0%	
Number of Slice LUTs	1	9,112	1%	
Number used as logic	1	9,112	1%	
Number using O6 output only	1			
Number using O5 output only	0			
Number using O5 and O6	0			
Number used as ROM	0			
Number used as Memory	0	2,176	0%	
Number of occupied Slices	1	2,278	1%	
Number of MUXCYs used	0	4,556	0%	
Number of LUT Flip Flop pairs used	1			
Number with an unused Flip Flop	1	1	100%	
Number with an unused LUT	0	1	0%	

Number of fully used LUT-FF pairs	0	1	0%	
Number of slice register sites lost to control set restrictions	0	18,224	0%	
Number of bonded IOBs	6	232	2%	
Number of RAMB16BWERs	0	32	0%	
Number of RAMB8BWERs	0	64	0%	
Number of BUFIO2/BUFIO2_2CLKs	0	32	0%	
Number of BUFIO2FB/BUFIO2FB_2CLKs	0	32	0%	
Number of BUFG/BUFGMUXs	0	16	0%	
Number of DCM/DCM_CLKGENs	0	4	0%	
Number of ILOGIC2/ISERDES2s	0	248	0%	
Number of IODELAY2/IODRP2/IODRP2_MCBs	0	248	0%	
Number of OLOGIC2/OSERDES2s	0	248	0%	
Number of BSCANs	0	4	0%	
Number of BUFHs	0	128	0%	
Number of BUFPLLs	0	8	0%	
Number of BUFPLL_MCBs	0	4	0%	
Number of DSP48A1s	0	32	0%	
Number of ICAPs	0	1	0%	
Number of MCBs	0	2	0%	
Number of PCILOGICSEs	0	2	0%	
Number of PLL_ADVs	0	2	0%	
Number of PMVs	0	1	0%	
Number of STARTUPs	0	1	0%	
Number of SUSPEND_SYNCs	0	1	0%	
Average Fanout of Non-Clock Nets	1.00			

<b>Performance Summary</b>	<b>[-]</b>		
<b>Final Timing Score:</b>	0 (Setup: 0, Hold: 0)	<b>Pinout Data:</b>	Pinout Report
<b>Routing Results:</b>	All Signals Completely Routed	<b>Clock Data:</b>	Clock Report
<b>Timing Constraints:</b>			

<b>Clock Report</b>	<b>[-]</b>
<b>Data Not Yet Available</b>	

<b>Detailed Reports</b>	<b>[-]</b>				
<b>Report Name</b>	<b>Status</b>	<b>Generated</b>	<b>Errors</b>	<b>Warnings</b>	<b>Infos</b>
Synthesis Report	Current	Mon Apr 20 06:23:02 2020	0	0	0
Translation Report	Current	Mon Apr 20 06:23:12 2020	0	0	0
Map Report	Current	Mon Apr 20 06:23:29 2020	0	0	6 Infos (0 new)
Place and Route Report	Current	Mon Apr 20 06:23:44 2020	0	0	2 Infos (0 new)
Power Report					
Post-PAR Static Timing Report	Current	Mon Apr 20 06:23:53 2020	0	0	4 Infos (0 new)
Bitgen Report	Out of Date	Sun Apr 19 11:12:05 2020	0	0	0

<b>Secondary Reports</b>	<b>[-]</b>	
<b>Report Name</b>	<b>Status</b>	<b>Generated</b>
ISIM Simulator Log	Out of Date	Mon Apr 20 06:22:38 2020
WebTalk Log File	Out of Date	Sun Apr 19 11:12:05 2020

**Date Generated:** 04/20/2020 - 06:23:55