Objective

- Learn and become familiar with Altera's Quartus II CAD SW & DE1 Cyclone II FPGA Board
- Learn how to design, simulate, synthesize, program on FPGA and test basic combinational digital components using Schematic design, Altera Quartus II CAD SW and DE1 FPGA board.
Instructions

1. The LAB#1 solution .pof and .sof files are provided to you. Your design must behave exactly same as this solution (except arbitrary group number, 353). You should use it as a golden reference during your design, testing & debugging your project. You must run this solution on your board and compare with your design whenever you have any question during your work.

2. You use schematic design for LAB#1. Your project name should be LxGzz where x=LAB number, zz=your group number. For example, a project name, L1G09, is a name of LAB#1 Project done by Group#9.

3. There are five(5) Parts in LAB#1. Each Part uses some or all of SWs {SW2, SW1, SW0}. Each Part, however, uses different outputs. See the following truth table as reference.

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>SW2 SW1 SW0</td>
<td>Part1(LEDG7) Part2(LEDG5) Part3(LEDG2) Part4(HEX3,2) Part5(HEX0)</td>
</tr>
<tr>
<td>0 0 0 (Down)</td>
<td>0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 1 (+Up)</td>
<td>1 1 1</td>
</tr>
<tr>
<td>0 1 0</td>
<td></td>
</tr>
<tr>
<td>0 1 1</td>
<td></td>
</tr>
<tr>
<td>1 0 0</td>
<td></td>
</tr>
<tr>
<td>1 0 1</td>
<td></td>
</tr>
<tr>
<td>1 1 0</td>
<td></td>
</tr>
<tr>
<td>1 1 1</td>
<td></td>
</tr>
</tbody>
</table>

4. Logic value [0,1] notation
On input SWs {SW2, SW1, SW0},
Down = 0; Up = 1.
On LED outputs,
OFF = 0; ON = 1.

5. Set tpd of LAB#1 as 8ns (you should set it by going to Assignment->Timing Analysis Setting-> Classic Timing Analyzer, and set it there)
6. A **functional** simulation waveform may be similar to following diagram
   (This is just an example output from LAB#1 solution, golden reference)

A **timing** simulation waveform may be similar to following diagram
Part 1 [3 points for Demo]:

3-WAY LIGHT CONTROLLER CIRCUIT DESIGN

An N-way light controller circuit changes the state of the output light (ON/OFF) whenever one input switch among N input switches changes (Up/Down) independent of other input switches. It is a very useful circuit for light control and is used widely in houses, buildings and many other places. Your house most likely has one at least.

Design a 3-way light controller circuit as follows.

Inputs: SW2, SW1, SW0
Output: LEDG7
Initial condition: LEDG7 is OFF when SW2=SW1=SW0=0 (i.e., all Down position)

Truth Table:

<table>
<thead>
<tr>
<th>SW2</th>
<th>SW1</th>
<th>SW0</th>
<th>LEDG7</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Initial condition</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Part 2 [3 points for Demo]:

2-TO-1 MUX DESIGN

An N-to-1 multiplexor (often referred to as "mux") takes N inputs and determines an output by selector. For example above 2-to-1 mux takes a and b inputs with s (selector). The output c is determined by selector's value as follows.

if s = 0 \[ c = a \]
if s = 1 \[ c = b \]

Design a 2-to-1 MUX circuit as follows.

Inputs: \( SW2, SW1, SW0 (=\text{selector}) \)
Output: \( LEDG5 \)
Operation:
\[
\begin{align*}
\text{if } SW0 &= 0 \quad \text{LEDG5} &= SW1 \\
\text{if } SW0 &= 1 \quad \text{LEDG5} &= SW2
\end{align*}
\]

Hint:
Figure out a truth table first.
Part 3 [3 points for Demo]:

**3-INPUT MAJORITY DETECTOR DESIGN**

A majority detector takes odd number of inputs, detects the majority of them, and make it output. If input contains more 1s than 0s, then output is 1. Otherwise output is 0.

For example, when there are five inputs,

output is 0 in case of "00000" input
output is 0 in case of "01001" input
output is 1 in case of "11001" input.
output is 1 in case of "11111" input.
extc.

Design a 3-input majority detector as follows.

Inputs: **SW2, SW1, SW0**
Output: **LEDG2**
Operation:

if the input contains more 1s than 0s (e.g., "111", "101", etc.)

**LEDG2** = ON (i.e., majority is 1)
else

**LEDG2** = OFF (i.e., majority is 0)

-------------------------------------------------------------------

**Hint:**

Figure out a truth table first.
Part 4 [3 points for Demo]:

GROUP NUMBER DISPLAY CIRCUIT DESIGN
*****************************************************

A Group Number Display circuit takes a switch input and displays your Group Number on 3-digit 7-segment displays, HEX3 HEX2 HEX1.

Design a Group Number Display circuit as follows.

Inputs:   SW0
Output:   HEX3, HEX2, HEX1
Operation:

if SW0 = 0  HEX3 HEX2 HEX1 displays "000".

if SW0 = 1  HEX3 HEX2 HEX1 displays your group number in 3-digit form.
For example, if your group number is 35, the output should be "35" as follows.

035

If your group number is single digit case (e.g., 7), then the output should be "007".
If your group number is three digits case (e.g., 107), then the output should be "107".

--------------------------------------------------

Hint:   See DE1 User manual sec. 4.3. for 7-segment display operation
Part 5 [3 points for Demo]:
BINARY-TO-DECIMAL DISPLAY DESIGN
******************************************************************************
A Binary-to-Decimal Display circuit converts a binary number to a decimal number and displays it on 7-segment HEX0 display. For example, if SW1 SW0 = "11" (=both Up), then '3' is displayed on HEX0.

Design a Binary-to-Decimal Display circuit as follows.

Inputs: SW1, SW0 (=binary number)
Output: HEX0
Operation:

HEX0 displays the decimal value of SW1 SW2 binary number. Your design should meet the following specifications.  (note: 0 for Down, 1 for Up position for SW)

if SW1 SW0 = 00 then '0' is displayed on HEX0
if SW1 SW0 = 01 then '1' is displayed on HEX0
if SW1 SW0 = 10 then '2' is displayed on HEX0
if SW1 SW0 = 11 then '3' is displayed on HEX0

For example, if SW1 SW0 = 11, HEX0 should be

[ ]

------------------------------------------------------------------------ Hint ------------------------------------------------------------------------

Hint: Part4 can be used as a reference for this part

------------------------------------------------------------------------ The End of LAB1 ------------------------------------------------------------------------