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. Your LAB#1 project name should be L1Cyyy, where yyy=your CID (e.g., L1C079 if your CID=079).

2. A golden solution .pof and .sof files are provided to you. You need to play with them as a reference during design whenever you have a question. Remember the following rules when using the golden solution.

   Rules when you use a golden solution:
   Your design does not need to behave same way as the golden solution behaves in all possible input conditions. However, your design MUST behave same way as the golden solution behaves in specific input condition(s) required in our LAB document. That is because we check ONLY such required specific input condition(s) and it’s expected result during Demo.

   For example: Suppose a LAB doc said "If SW0 is up, turn LED0 on." and no further specifications on SW0.

   In this case, we check only the specific input condition (i.e., SW0 is up) and it’s expected result (i.e., LED0 is on). Therefore your design must follow the golden solution behavior in this specific input condition (SW0 is up) and it’s result (LED0 is on).

   All other input conditions (e.g., SW0 is down, ... etc.) and it’s result will not be checked during Demo, so your design does not need to follow the golden solution behavior in other input conditions (i.e., you will not receive any point deduction even though your design behaves differently from golden solution in other input conditions).

3. Use schematic design. The LAB#1 consists of five(5) Parts, Part 1 - 5. Each Part uses some or all of switches, {SW2, SW1, SW0}. Each Part has different output. See the following diagram as a reference.

   Note: Part 1 is a prerequisite for other Parts. You will get zero(0) point for LAB#1 if Part 1 fails.

4. You should set tpd for LAB#1 as 8ns

   Hint: You can set this by going to Assignment->Timing Analysis Setting->Classic Timing Analyzer, and set it there.
5. **Functional vs. Timing Simulation example** (Following diagrams are example purpose only, NOT your solution!)

An example **functional** simulation waveform may be similar to following diagram

![Functional Simulation Waveform](image1)

An example **timing** simulation waveform may be similar to following diagram

![Timing Simulation Waveform](image2)
Part 1
YOUR CLASS ID(CID) DISPLAY CIRCUIT DESIGN

**************************************************
( Note: Part1 is a prerequisite for all other Parts. You will get zero(0) point for LAB#1 if Part1 fails. )

A CID display circuit takes a switch(SW) input and displays your CID on 3-digit 7-segment displays, \{HEX3
HEX2 HEX1\}.
Design a CID Display circuit as follows.

Inputs: \textbf{SW0}
Output: \textbf{HEX3, HEX2, HEX1} \hspace{1cm} // HEX0 is not used in Part1

Operation:

\begin{itemize}
  \item if \textbf{SW0} = 0\,(i.e., \text{down}) \hspace{1cm} \text{HEX3 HEX2 HEX1} \text{ displays "000".}
  \item if \textbf{SW0} = 1\,(i.e., \text{up}) \hspace{1cm} \text{HEX3 HEX2 HEX1} \text{ displays your CID in 3-digit form.}
\end{itemize}

\textbf{Ref:} The golden solution displays "353"(since it's CID was set to 353 – a CID that no student will use).

************************************************** The End of Part1 **************************************************

---------------------------------------- \textit{Hints} ----------------------------------------

For example, if your CID is 035, the output should be "035" displayed as follows.

\[035\]

Suppose your class ID is 7, then the output should be "007".
If your class ID is 247, then the output should be "247".

Study DE1 User manual sec. 4.3. for 7-segment display operation carefully. It will help you a lot.
**Part 2**  
**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 position 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: \( SW_2, SW_1, SW_0 \)  
Output: \( LEDR_1 \)  
Initial condition: \( LEDR_1 \) is OFF when \( SW_2=SW_1=SW_0=0 \) (i.e., all down position)

Truth Table:

<table>
<thead>
<tr>
<th>SW2</th>
<th>SW1</th>
<th>SW0</th>
<th>LEDR1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0 (OFF) &lt;--- Initial condition</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1 (ON)</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>

**********************************************************************  The End of Part2  **********************************************************************
2-TO-1 MUX DESIGN

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

\[
\begin{align*}
\text{if } s = 0 & \quad c = a \\
\text{else (i.e., } s = 1) & \quad c = b
\end{align*}
\]

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

Inputs: \( SW2, SW1, SW0 \) (=selector)
Output: \( LEDR4 \)
Operation:

\[
\begin{align*}
\text{if } SW0 = 0 & \quad LEDR4 = SW1 \\
\text{else (i.e., if } SW0 = 1) & \quad LEDR4 = SW2
\end{align*}
\]

***************  The End of Part3  ***********************
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.

Design a 3-input majority detector as follows.

Inputs: $SW2, SW1, SW0$
Output: $LEDR8$
Operation:
   if the majority is 1
      $LEDR8 = ON$
   else // i.e., the majority is 0
      $LEDR8 = OFF$

----------------------- The End of Part 4 -----------------------

----------------------- Hints -----------------------
For example,
   if the input = 111 => the majority is 1 => $LEDR8 = ON$
   if the input = 101 => the majority is 1 => $LEDR8 = ON$
   ..
   if the input = 010 => the majority is 0 => $LEDR8 = OFF$
   if the input = 000 => the majority is 0 => $LEDR8 = OFF$
Part 5
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" (in binary), then '3' is displayed on HEX0.

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

Inputs:   SW1, SW0 (in binary) where, SW0 is LSB (Least Significant Bit)
Output:   HEX0 (in decimal)

Operation:

HEX0 displays the decimal value of SW1 SW0 (binary number). Your design should meet the following specifications.

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

3

The End of Part5