Objective

- Based on the experience gained from LAB#1, learn how to design, simulate, synthesize, program on FPGA and test combinational & sequential digital components using Altera Quartus II CAD SW and DE1 FPGA board.
- Learn and become familiar with logic design using Verilog Hardware Description Language
Instructions

1. The solution .pof and .sof files are provided to you. Your design must behave exactly same as this solution (except blinking ledg[0] & GN=99 which are a solution identifier). You should use the given solution 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. Use Verilog HDL. Like previous project, your project name should be LxGzz where x=LAB number, zz=your group number. For example, a project name, L2G09, is a name of LAB#2 Project done by Group#9.

3. Use following top-level module interface code in your design. No part of this code is allowed to be modified.

   Hint: The top-level module name must be same as the project name (e.g., L2G09) in our Quartus.

   ```
   module LxGzz(
     input [9:0] sw, // ten up-down switches, SW9 - SW0
     input [3:0] key, // four pushbutton switches, KEY3 - KEY0
     input clock, // 24MHz clock source on Altera DE1 board
     output [9:0] ledr, // ten Red LEDs, LEDR9 - LEDR0
     output [7:0] ledg, // eight Green LEDs, LEDG8 - LEDG0
     output reg [6:0] hex3, hex2, hex1, hex0 // four 7-segment, HEX3 - HEX0
   );
   ```

4. New rules for this LAB
   4.1 In the LAB Report, replace Functional & timing simulation waveforms with a Compilation Report - Flow Summary" of your design. Clearly mark the Percentage(%) value of "Total logic elements".
   4.2 No tpd timing setup is required.

5. Our acceptable timing margin(or tolerance) for real-time operation is -30 and +30%.
   For example, in case of 1 second required in Part4&5 of this LAB, a time period between 0.7 sec(= -30%) and 1.3 sec(= +30%) is acceptable as 1 sec period. A time period beyond this range is an error and will get zero point.

6. Use of simulation for debugging
   Simulation is one of the most effective debugging methods in digital design engineering. Like many professional engineers, you have to use simulation to figure out the cause of the problem when your design behaves differently than you expect. Without simulation, you may have difficulties in solving your problem and waste your time.
7. LAB Project Operation

7.1 Initial Power-on State

When DE1 board's power is turned on, following conditions must be implemented:
- all sw are in DOWN position
- all key are NOT PRESSED
- all ledg and ledr are OFF
- hex[3:0] displays "140L"

CAUTION: This is a prerequisite condition. You will get zero(0) point for this LAB if you fail this condition.

7.2 Operation (Reminder: Check with the solution given to you whenever necessary for complete details.)

In this LAB, the sw[9:5] is an enable selector of different Parts of the LAB. You enable a particular Part by changing the sw[9:5] as follows. Note that only one Part is enabled at a time during operation.

IF sw[9:5]=00000 // NO sw is in UP position
    NO Part is enabled and all ledg and ledr are OFF and hex[3:0] displays "140L"

IF sw[9:5]=10000 // only sw[9] is in UP position
    Part1 is enabled AND all other Parts are disabled

IF sw[9:5]=01000 // only sw[8] is in UP position
    Part2 is enabled AND all other Parts are disabled

IF sw[9:5]=00100 // only sw[7] is in UP position
    Part3 is enabled AND all other Parts are disabled

    Part4 is enabled AND all other Parts are disabled

    Part5 is enabled AND all other Parts are disabled
PART 1 [3 points for Demo]:
Your GN (Group Number) Display design

**************************************************************

IF sw[9:5]=10000 // only sw[9] is in UP position
                Part 1 is enabled AND all other Parts are disabled

Design a Group Number Display circuit as follows.

Inputs:     SW[9]  
Output:     HEX[3:0]  
Operation (Reminder: Check with the solution given to you whenever necessary for complete details.)

if SW[9] = 0 see Sec. 7. LAB Project Operation of Instructions
if SW[9] = 1 perform following operation.

    HEX[3:0] displays your group number in 4-digit form. For example, if your group number is 35, the
    HEX[3:0] displays "0035" as follows.

    0035

    If your group number is 7, then the output should be "0007".
    If your group number is 107, then the output should be "0107".

-------------------------------------------------------- Hints --------------------------------------------------------

Hint: See DE1 User manual sec. 4.3. for 7-segment display operation
PART 2 [3 points for Demo]:
Decimal Multiplier Calculator design

IF sw[9:5]=01000  // only sw[8] is in UP position
Part2 is enabled AND all other Parts are disabled

Design a Decimal Multiplier circuit as follows.

Inputs:  
op1 is HEX[3] which is a decimal value of SW[4:3],
op2 is HEX[2] which is a decimal value of SW[2:1],
HEX[1] should be OFF (i.e., No light on HEX[1])

Output:  
HEX[0] is the decimal value of multiplication result

Operation:  (Reminder:  *Check with the solution given to you whenever necessary for complete details.* )
if SW[8] = 0  see  Sec. 7.  LAB Project Operation of Instructions.
if SW[8] = 1  perform following operation.


For example,

IF  SW[4:3]= 00  AND  SW[2:1]=00  THEN  HEX[3:0] = 00  0
IF  SW[4:3]= 01  AND  SW[2:1]=00  THEN  HEX[3:0] = 10  0
IF  SW[4:3]= 00  AND  SW[2:1]=01  THEN  HEX[3:0] = 01  0
etc.
PART 3 [3 points for Demo]:

Modulo-10 Up counter design

********************************************************************************

IF sw[9:5]=00100  // only sw[7] is in UP position
    Part3 is enabled AND all other Parts are disabled

Design a Modulo-10 Up Counter circuit as follows.

Inputs:  KEY[3] for pressing input,
        SW[0] for reset operation

Output:  HEX[3:0] for counter output

Operation: (Reminder: Check with the solution given to you whenever necessary for complete details.)
    if SW[7] = 0  see  Sec. 7. LAB Project Operation of Instructions.
    if SW[7] = 1  perform following operation.
        1) Your circuit counts the number of pressing on KEY[3] and displays the result on HEX[3:0]. The counter output increases by one each time KEY3 is pressed (and should NOT change when KEY[3] is released).
        2) When the output=9 and KEY[3] is pressed, the output is cleared to 0 (=modulo-10 operation)
        3) SW[0] is a reset switch. If SW[0]=1 then the output is cleared to 0 immediately. The counter operates normally when SW[0]=0.

Example:

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0 (=no pressing)</td>
<td>0000</td>
</tr>
<tr>
<td>1 (=first pressing)</td>
<td>0001</td>
</tr>
<tr>
<td>2 (=second pressing)</td>
<td>0002</td>
</tr>
<tr>
<td>3</td>
<td>0003</td>
</tr>
<tr>
<td>4</td>
<td>0004</td>
</tr>
<tr>
<td>5</td>
<td>0005</td>
</tr>
<tr>
<td>6</td>
<td>0006</td>
</tr>
<tr>
<td>7</td>
<td>0007</td>
</tr>
<tr>
<td>8</td>
<td>0008</td>
</tr>
<tr>
<td>9</td>
<td>0009</td>
</tr>
<tr>
<td>10</td>
<td>0000</td>
</tr>
<tr>
<td>11</td>
<td>0001</td>
</tr>
<tr>
<td>12</td>
<td>0002</td>
</tr>
<tr>
<td>13</td>
<td>0003</td>
</tr>
<tr>
<td>....</td>
<td>....</td>
</tr>
</tbody>
</table>
PART 4 [3 points for Demo]:
Real-time 1-second period clock signal timer design

****************************************************************************


Part4 is enabled AND all other Parts are disabled

Design a real-time 1-second period clock signal timer circuit as follows.

**Inputs:**  SW[6]
**Output:**  HEX[3:0] for timer output
            LEDG[7] for blinking signal

**Operation** (Reminder: Check with the solution given to you whenever necessary for complete details.)

if SW[6] = 0  see  Sec. 7. LAB Project Operation of Instructions
if SW[6] = 1  perform following two operations.

1. HEX[3:0] starts displaying continuously the number of seconds passed since SW[6] goes up. For example, if 35 seconds passed since SW[6] went up, the HEX[3:0] output should be "0035" as follows.

   0035

2. The LEDG[7] starts blinking every second with 50% duty cycle as follows.
PART 5 [3 points for Demo]:
Bouncing Ball design

*****************************************************************************

Part5 is enabled AND all other Parts are disabled

Design a bouncing ball (one red LED light) circuit as follows.

Inputs:  
SW[5]
SW[0] for pausing

Output:  
LEDR[9:0] for bouncing ball movement
HEX[3:0] should be all OFF (i.e., No light at all)

Operation (Reminder: Check with the solution given to you whenever necessary for complete details.)

if SW[5] = 0  see  Sec. 7. LAB Project Operation of Instructions
if SW[5] = 1  perform following operation.

Starting from LEDR[0] position, the red light ball moves from LEDR[0] to LEDR[9] with a duration of 0.5 second. When arrived at LEDR[9], the ball moves from LEDR[9] back to LEDR[0] with same duration of 0.5 second. Therefore the time period of one round trip is one second. When returned to LEDR[0], the ball keeps repeating the same movement.

Pause switch: SW[0]
SW[0] = 1 pauses the operation.
SW[0] = 0 allows the operation.

--------------- The End of LAB2 ------------------------