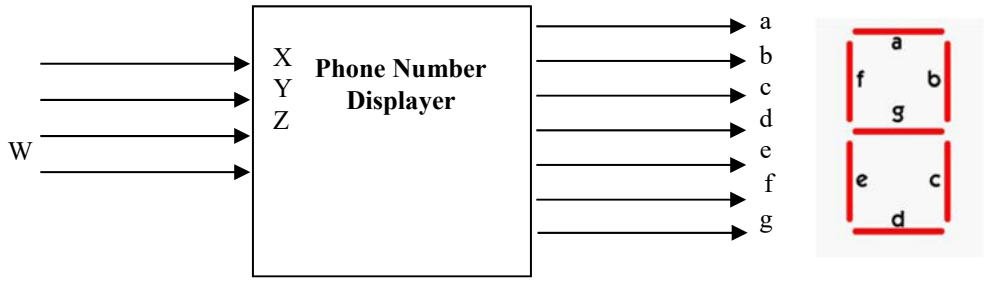
Technical Objective:

Prior to the introduction of hardware description languages (HDL), digital circuits were designed using basic logic gates. Logic functions were reduced to their simplest form and implemented using a library of parts. Circuit design using HDL has replaced this process. However, the tools that synthesize the HDL still reduce the circuit to its simplest logic equation.

The purpose of this lab is to manually reduce a complex logic equation and implement it given a limited set of logic gates using schematic capture. Simulation will be used to verify correct functionality of the reduced function.

Pre-Laboratory: (30% of the grade)

1. The block diagram shown below represents the **Phone Number Displayer** that has four inputs and seven outputs. The four inputs, X, Y, Z and W select which digit of the phone number is displayed. The seven outputs, a, b, c, d, e, f and g drive the segments of a common anode seven segment display.



2. For the phone number 585-475-6609, the seven segment display will be driven as follows:

X	Y	Z	W	7-Segment Display	a	b	c	d	e	f	g
0	0	0	0	5	0	1	0	0	1	0	0
0	0	0	1	8	:	:	:	:	:	:	:
0	0	1	0	5	:	:	:	:	:	:	:
0	0	1	1	-	1	1	1	1	1	1	0
0	1	0	0	4	:	:	:	:	:	:	:
0	1	0	1	7	:	:	:	:	:	:	:
0	1	1	0	5	:	:	:	:	:	:	:
0	1	1	1	-	:	:	:	:	:	:	:
1	0	0	0	6	:	:	:	:	:	:	:
1	0	0	1	6	:	:	:	:	:	:	:
1	0	1	0	0	:	:	:	:	:	:	:
1	0	1	1	9	:	:	:	:	:	:	:
1	1	0	0	blank	1	1	1	1	1	1	1
1	1	0	1	blank	1	1	1	1	1	1	1
1	1	1	0	blank	1	1	1	1	1	1	1
1	1	1	1	blank	1	1	1	1	1	1	1

3. Complete the truth table for your own phone number

4. Using k-maps, Write the <u>minimal</u> sum-of-products expression for the seven outputs.

5. Design your circuit using standard SSI gates. Standard SSI gates include AND, OR, NAND, NOR, XOR, XNOR and NOT.

6. Capture your circuit using the schematic editor in Quartus Prime.

7. Submit the following **prior** to your lab session. You may put them in the drop box or bring them to lab
    - All design work including truth tables and k-maps
    - Your Quartus Prime schematic

Procedure: (70% of the grade)

1. Create a simulation input wave file that covers all input combinations. It may be easier to define each input signal as a clock since each will transition at a regular period.

   To do so, click on the clock icon  in the waveform editor and then specify a period. For example, W (lsb) needs to toggle every 10 ns and will have a period of 20 ns. The next significant bit, Z, will toggle every 20 ns and have a period of 40 ns, the next bit will have a period of 80 ns, and so on.
2. Run the simulation long enough to test all 16 combinations of inputs and verify that the outputs are active for the appropriate input values and are not active for any other input values.
3. If the output is not correct, troubleshoot the design.
4. Obtain a signoff when the simulation runs correctly.
5. Annotate comments onto the output waveform to show how you know the circuit is correct.
6. Using the DE0-CV pinout document found in MyCourses, assign the four inputs to SW0-SW3 and the seven outputs to the HEX0 seven segment display (see lecture 2 for more information).
7. Recompile and program the DE0-CV board.
8. Once you have verified you circuit is operating correctly, obtain a signoff for a working board.

Documentation:

There is no documentation due next week. Please save all of your design work, and your simulation results as they will become part of your documentation packet for lab 4.

Signoffs and Grade:

Please submit with Lab 4 Report

Name:_____

Component	Signoff	Date	Time
Functional Simulation (35 pts)			
Working Board (35 pts)			

==========

Component	Received	Possible
Prelab		30
Signoff		70
Penalties • after the first 15 minutes of lab session 3: -10 • after the first 15 minutes of lab session 4: -25 • no signoffs after the first 15 minutes of lab session 5:	-	
Total		100