Name: _____

ECET 331 – Digital Integrated Circuits

Lab 5
BCD Adder

Objective: Students successfully completing this lab exercise will accomplish the following objectives:

1. Augment their experience in using Quartus II to enter, simulate and implement a digital design.
2. Become familiar with the concept of a BCD adder.
3. Augment their experience with Karnaugh maps to include maps with five input variables.
4. Gain experience using VHDL to describe the BCD adder design in both a Boolean expression format and in a truth table format.

Equipment: Laptop PC running Quartus II, Altera UP-2 trainer board and cabling, jumper wires.

Procedure 1 - Schematic Entry:

1. Create a folder on the T: drive of your computer or your portable storage media called "*yourname*_lab5". Also create a corresponding project by the same name. Create new folders as you study new circuits.

2. The truth table for a BCD adder conversion circuit is shown below. Karnaugh maps are also included with this lab. Study section 7.7 in the textbook and use the K-maps to generate logic expressions for the five outputs.

Truth Table: Modified from Table 7.11, page 389 of the text.

Binary Sum of Two BCD Digits + Carry In Bit					Decimal	Corrected BCD (Carry Out + BCD)	
Cin	S_3	S_2	S_1	S_0		C_{OUT}	$T_3T_2T_1T_0$
0	0	0	0	0	0	0	0000
0	0	0	0	1	1	0	0001
0	0	0	1	0	2	0	0010
0	0	0	1	1	3	0	0011
0	0	1	0	0	4	0	0100
0	0	1	0	1	5	0	0101
0	0	1	1	0	6	0	0110
0	0	1	1	1	7	0	0111
0	1	0	0	0	8	0	1000
0	1	0	0	1	9	0	1001
0	1	0	1	0	10	1	0000
0	1	0	1	1	11	1	0001
0	1	1	0	0	12	1	0010
0	1	1	0	1	13	1	0011
0	1	1	1	0	14	1	0100
0	1	1	1	1	15	1	0101
1	0	0	0	0	16	1	0110
1	0	0	0	1	17	1	0111
1	0	0	1	0	18	1	1000
1	0	0	1	1	19	1	1001

3. Generate and enter schematics for the output expressions from step 2.

4. Compile your design. Assign the appropriate device for your Altera UP-2 board. Assign pin numbers and recompile.

5. Simulate your design. The five inputs (C, S_3, S_2, S_1 and S_0) can be combined as a group and the truth table below can be displayed in time as we saw in a previous lab. Verify that your circuit performs correctly. Investigate and fix any errors before proceeding to the next step.

6. Have the instructor verify your working design in simulation. Turn in a copy of your schematic and a copy of your waveform output for input combinations 0-31 (either printed from the simulation or hand-drawn). Explain any unexpected outputs.

Instructor's Signature: _____ Date: _____

Procedure 2 - VHDL Boolean Expression Entry:

7. Create a separate folder on the T: drive of your computer or your portable storage media called "*yourname*_lab5_be" (for "Boolean Expression"). Also create a corresponding project by the same name.

8. In the project, open a new VHDL file. Save the file as "*yourname*_lab5_be.vhd". Use the Quartus II text editor to enter your design as described below.

9. Use pages 133-140 in your textbook as a guide to writing VHDL. Create a VHDL representation of your design by writing your design expressions as concurrent statements in the VHDL language. Take notice of syntax conventions and order of operations.

   Note: The name of the entity is given in the first and last lines of the entity declaration and ***must*** match the file name.

10. Compile and simulate your design as you did in steps 4 and 5 above.

11. Have the instructor verify your working design in simulation. Turn in a copy of your source code and simulation output for input combinations 0-31.

Instructor's Signature: _____ Date: _____

Procedure 3 - VHDL Truth Table Entry:

12. Create a separate folder on the T: drive of your computer or your portable storage media called "*yourname*_lab5_tt" (for "Truth Table"). Also create a corresponding project by the same name.

13. In the project, open a new VHDL file. Save the file as "*yourname*_lab5_tt.vhd". Use the Quartus II text editor to enter your design as described below.

14. A truth table is encoded in VHDL using a selected signal assignment statement. An example of this construct is shown in your textbook on pages 138-9.

    Another example (for a BCD-to-Seven-Segement Decoder) is shown on the last page of this lab handout. Note how the input and output ports are converted to bit_vectors for use in the architecture part of the file. This example is discussed in the textbook on pages 285-6

15. Compile and simulate your design as before. Have the instructor verify your working design in simulation.

16. Have the instructor verify your working designs in simulation. Turn in a copy of your source code and simulation output for input combinations 0-31.

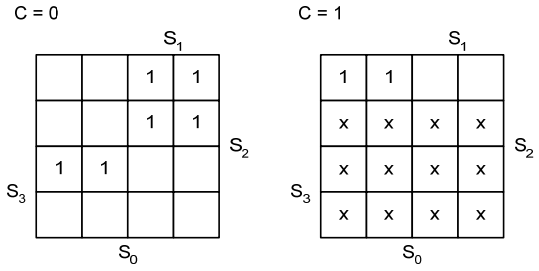
Instructor's Signature: _____ Date: _____

Karnaugh Maps:

1. For T_0: By inspection.

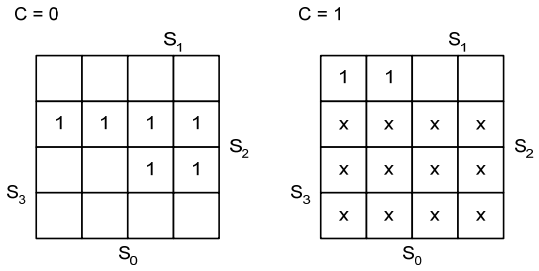
$$T_0 = S_0$$

2. For T_1: Make sure you are able to derive the given equation.

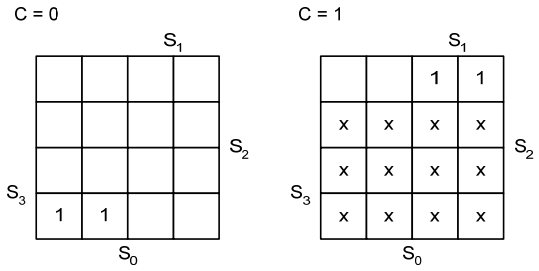


$$T_1 = \overline{C}\,\overline{S_3}S_1 + S_3S_2\,\overline{S_1} + C\overline{S_1}$$

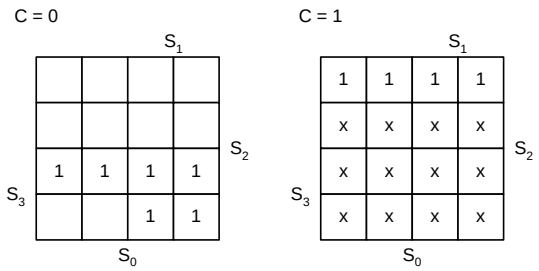
3. For T_2: Derive the equation for T_2 from the K-map below.



4. For T_3: Derive the equation for T_3 from the K-map below.



5. For C_{OUT}: Make sure you are able to derive the given equation



$$C_{OUT} = C + S_3S_2 + S_3S_1$$

Coding Example:

```
-- bcd_7seg.vhd
-- BCD-to-seven-segment decoder
-- Programmed by Bob Dueck

ENTITY bcd_7seg IS
       PORT(
              d3, d2, d1, d0         : IN    BIT;
              a, b, c, d, e, f, g    : OUT   BIT);
END bcd_7seg;

ARCHITECTURE seven_segment OF bcd_7seg IS
       SIGNAL input : BIT_VECTOR (3 DOWNTO 0);
       SIGNAL output: BIT_VECTOR (6 DOWNTO 0);
BEGIN
       input <= d3 & d2 & d1 & d0;
       WITH input SELECT
              output <=      "0000001" WHEN "0000",
                             "1001111" WHEN "0001",
                             "0010010" WHEN "0010",
                             "0000110" WHEN "0011",

                             "1001100" WHEN "0100",
                             "0100100" WHEN "0101",
                             "1100000" WHEN "0110",
                             "0001111" WHEN "0111",

                             "0000000" WHEN "1000",
                             "0001100" WHEN "1001",
                             "1111111" WHEN OTHERS;

              -- Separate the output vector to make individual pin outputs.
              a       <=     output(6);
              b       <=     output(5);
              c       <=     output(4);
              d       <=     output(3);
              e       <=     output(2);
              f       <=     output(1);
              g       <=     output(0);

END seven_segment;
```

Extra Credit (20 points):

Add additional code to either of the VHDL forms of your BCD adder design to activate the two seven-segment displays on the Altera UP-2 board. Program the board. Have the instructor verify your working circuit.

Instructor's Signature: _____ Date: _____