Fundamentals of Digital System Design

ECE/CS 3700
Spring 2015, Homework # 2
Assigned, Wednesday Feb 4, Due Date: Wednesday, Feb 11, 2015, by 5pm in the HW locker.

1) **(FPGA Placement and Routing - 10 points)** Refer to Fig. B.39, in the textbook, given on page 768 Appendix B. The figure shows how Look-up Tables (LUTs) are interconnected by two layers of wiring: horizontal and vertical. The *blue coloured* cross-mark ($\times$) indicates that a connection *has been made* between the horizontally and vertically drawn wires. Convince yourselves that $f = f_1 + f_2 = x_1 x_2 + x_2' x_3$.
Based on the above concepts, you are asked to *program* an FPGA, whose LUTs and inter-connection wires are shown in Fig. 1. The function to be implemented is $f = f_1 \cdot f_2$, where $f_1 = a + \overline{b}$ and $f_2 = \overline{a} + \overline{c}$. LUT 1 should implement f_1. LUT 2 should implement f_2 and LUT 3 should implement $f_1 \cdot f_2$. The horizontally and vertically placed interconnection wires are fabricated in different planes. In order to depict a connection between these wires at a cross-point, place a cross-mark ($\times$). The inputs a, b, c and the output f have already been connected to the "input-output pads" for your (in)convenience. Have fun!

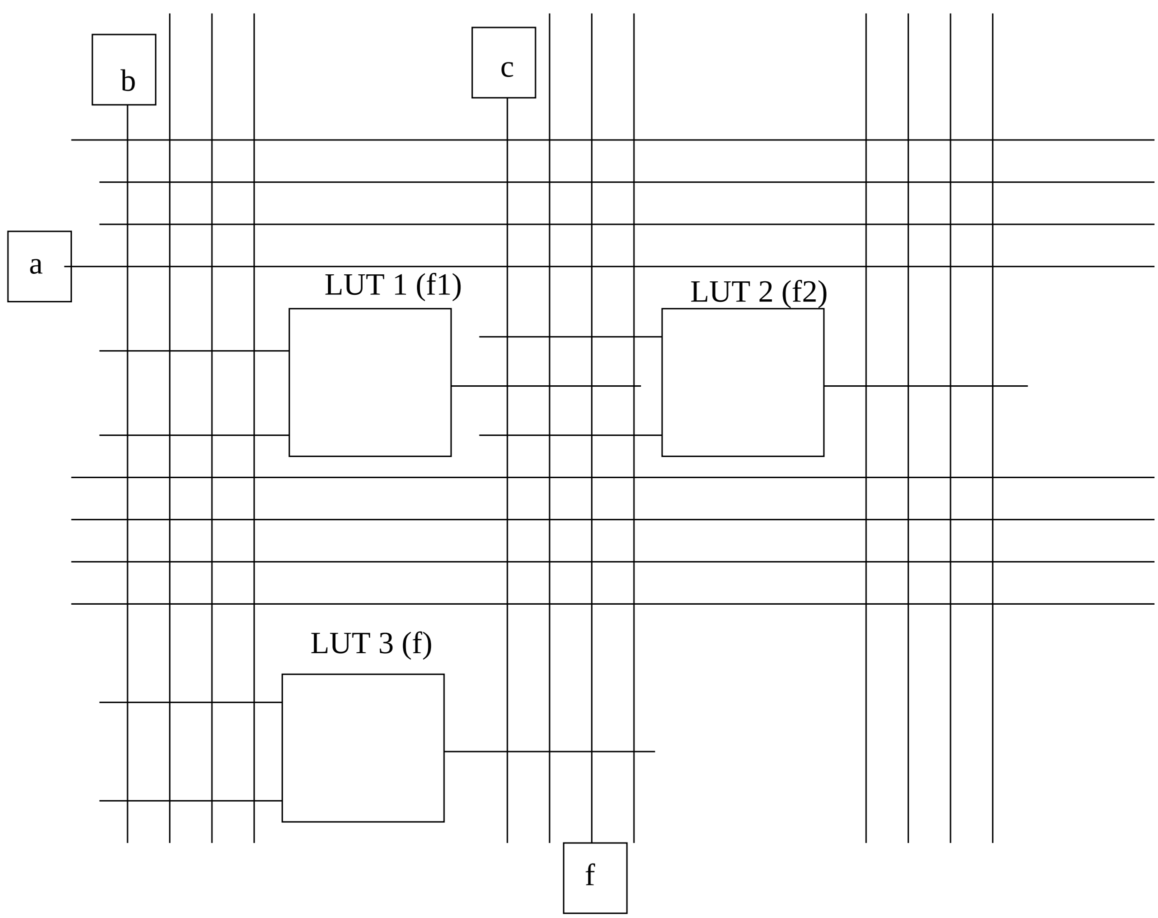

Fig. 1. Fill-up the truth-table entries in the look-up tables. Label the inputs and outputs of each LUT properly. Put a $\times$ mark to show an electrical connection between the vertical and horizontal metal wires.

2) **(MUX-based design - 10 points)** Given a Boolean function $F(a, b, c) = a \oplus b \oplus c$, implement it using *only* multiplexor (MUX) gates and $0, 1$ inputs. How many MUXes do you need? [Note: I covered design of Boolean functions using MUXes in class lectures].

3) **(K-Map minimization - 25 points)** For the following functions, whose on-set minterms are shown using the $sigma(\sum)$ notation, derive a minimum Sum-of-Product (SOP) form expression using Karnaugh maps (K-maps). Note that your final answer should be a sum-of-product form Boolean expression, derived using cube-covering on the K-maps.
   - $F(A, B, C, D) = \sum m(0, 2, 3, 5, 6, 7, 8, 10, 11, 14, 15)$
   - $F(A, B, C, D) = \sum m(1, 2, 3, 6, 7, 11)$
   - $F(A, B, C, D) = \sum m(2, 3, 5, 7, 10, 11, 13, 14, 15)$
   - $F(A, B, C, D, E) = \sum m(2, 5, 7, 8, 10, 13, 15, 17, 19, 21, 23, 24, 29, 31)$
   - $F(A, B, C, D, E) = \sum m(0, 4, 18, 19, 22, 23, 25, 29)$

4) **(K-maps with *don't cares* - 10 points)**. Derive minimum cost SOP forms for these functions.
   - $F(A, B, C, D) = \sum m(1, 3, 5, 7, 9) + \sum d(6, 8, 12, 13)$
   - $F(A, B, C, D) = \sum m(0, 2, 8, 9, 10, 15) + \sum d(1, 3, 6, 7)$

5) **(Another Min cost SOP - 15 points)** Consider the function $f(x_1, \ldots, x_4) = \sum m(0, 3, 4, 5, 7, 9, 11) + D(8, 12, 13, 14)$. Working on a K-map, first generate and list all the *prime implicants* of the function. Subsequently, from among these primes, identify the essential primes, and then derive a *minimum (literal) cost* SOP form Boolean expression. How many product-terms does the min-cost SOP form have? What is the total literal cost of the min-cost SOP?