

University of Cincinnati
Department of Electrical Engineering and Computing Systems
EECE 2060C – Digital Design, Lab room 806/808 Rhodes, Section# 005
Andrew Koski, M14561095, 9/12/2022

Lab 3: Computer Interface Design

Design Specification

A computer interface to a line printer has seven data lines that control the movement of the paper and the print head and determine which character to print. The data lines are labeled A, B, C, D, E, F, and G, and each represents a binary 0 or 1. When the data lines are interpreted as a 7-bit binary number with line A being the most significant bit, the data lines can represent the numbers 0 to $(127)_{10}$. The number 1310 is the command to return the print head to the beginning of a line, the number 1010 means to advance the paper by one line, and the numbers $(32)_{10}$ to $(127)_{10}$ represent printing characters. Details can be found in the ASCII code table attached. Design a digital circuit with 7 inputs A to G and 3 outputs X, Y, Z with the following specifications: (a) When the data lines A to G indicate a command (by the computer) to return the print head to the beginning of the line, X outputs logic 1; otherwise, X outputs logic 0. (b) When the data lines A to G indicate a command (by the computer) to advance the paper by one line, Y outputs logic 1; otherwise, Y outputs logic 0. (c) When the data lines A to G indicate a printable character (by the computer), Z outputs logic 1; otherwise, Z outputs logic 0. Inputs A, B, C, ..., G are implemented by switches and outputs X, Y, Z are implemented using LEDs.

Task 1 – Circuit Design and Implementation

Follow the design steps discussed in the classes. First, you design switching expressions for outputs X, Y, and Z. Unfortunately, there are 7 inputs, and it is not feasible to use a truth table to represent your design. So, use your logical thinking to write down the switching expressions for X, Y, and Z. Second, simplify the switching expressions as much as possible for outputs X, Y, and Z, using Boolean algebra you have learned so far. Keep in mind that your logic gates for your breadboard can only handle a max of four inputs (for AND/OR gates). Third, convert the switching expressions into gate-level circuits. Make sure you have the corresponding gates in your lab kit to implement the equations. For example, you cannot implement $N=A.B.C.D.E.F$ since there is no AND gate with 6 inputs in our lab kits. Points will be taken off if you use non-existent gates in your lab kits. Finally, implement the circuit using gates on breadboard.

Boolean Algebra and logic-

1

A B C D E F G

From 32_{10} to 127_{10} for possible decoder

$32_{10} = 32 \cdot \frac{16}{16} = \frac{512}{16}$ $127_{10} = 127 \cdot \frac{1}{1} = \frac{127}{1}$

$32_{10} = (100000)_2$
 $127_{10} = (111111)_2$

$(A'+A) \cdot (b) \cdot (C+C) \cdot (d'+d) \cdot (e'+e) \cdot (F'+F) \cdot (G'+G) =$

So from 32_{10} to 127_{10}
 $1 \cdot B \cdot 1 \cdot 1 \cdot 1 \cdot 1 \cdot 1 = \boxed{B} = 2$

From 1010100 , or $1101 = 13_{10}$
 $(A \cdot B' \cdot C \cdot D' \cdot E(F'+F)(G'+G)) = A \cdot B' \cdot C \cdot D' \cdot E = \lambda$

For 0001010 , or $1010 = 10_{10}$
 $(A'+A)(b)(e') \cdot d' \cdot e' \cdot G' = b' \cdot d' \cdot e' \cdot F' \cdot G' = \gamma$

minimized

$X'Y'Z' + X'YZ' + X'YZ'$

so $(A \cdot B' \cdot C \cdot D' \cdot E) + b' \cdot d' \cdot e' \cdot F' \cdot G' + \boxed{b}$

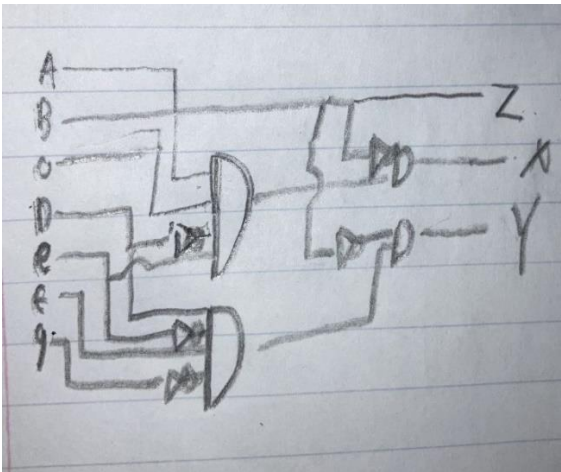
$b'(A \cdot C \cdot D' \cdot E + d' \cdot e' \cdot F' \cdot G') + b$

0001/11 10

X	Y	Z	1	0
0	0	1	0	1
1	0	0	0	0

$X'Y'Z'$

Circuit-



Overall Test Cases-

Inputs							Outputs			Purpose
A	B	C	D	E	F	G	X	Y	Z	
1	0	1	0	1	0	0	0	1	0	This circuit implementation is to find just Y
0	0	0	1	0	1	0	1	0	0	This circuit implementation is to find just X
0	0	0	0	0	0	0	0	0	0	This circuit implementation is to find NOT(X+Y+Z)
1	0	1	1	1	1	1	0	0	0	This circuit implementation is to find NOT(X+Y+Z)
0	0	0	0	0	0	0	0	0	0	This circuit implementation is to find NOT(X+Y+Z)
0	1	0	0	0	0	0	0	0	1	This circuit implementation is to find just Z
1	1	1	1	1	1	1	0	0	1	This circuit implementation is to find just Z
1	1	1	0	1	0	0	0	0	1	This circuit implementation is to find just Z against Y
1	1	0	1	0	1	0	0	0	1	This circuit implementation is to find just Z against X

Explanation-

The circuit has some shortcomings that we can exploit, starting with what is classified as printable character. The circuit in the end only needs to detect if the binary number representation is greater than 0100000. Since this is B in our binary representation, we only need this number to have it only output B as Z. This way we only need two 4 input and gates with inverters. So, we can represent 1101 as 1010100. Reason why it is not 0110100 is due to b in our binary representation being 100000. The other 4 input and gate represent 1010 or in our 7-input binary, 0001010. This results in D and E representing and overlapping both 4 input logic gates. To make sure that the combination 100000 and either 0001010 or 1010100 we add two and gates with inverters connected to B the circuit only allows X or Y to be 1 when b is not on.

Time Taken:

3 hours 30 minutes