Lab 5

EL 114 Digital Logic Design

Notes:

- In your lab-book, remember to write your steps/methods, and the observations/results
- Get TA's signature after completing each question.

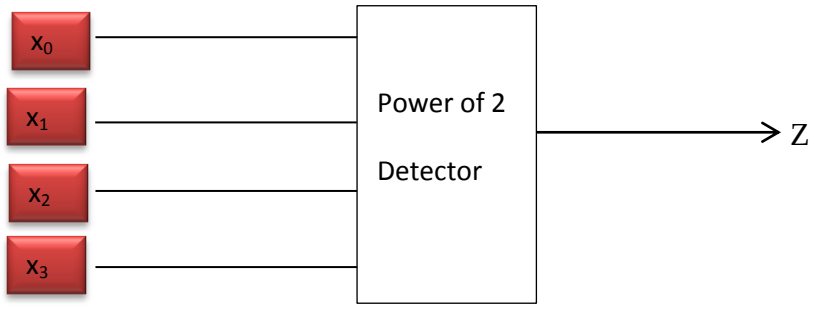
- All the following questions are to be done in Logisim.
- Make sure you draw each new schematic using "Add circuit" option in the Project menu, so that you can later make use of the instances of that particular schematic/circuit to build larger schematics.

1. Design a combinational circuit that accepts a 2-bit number and generates a 4-bit binary number output equal to the square of the input number.
2. Design a combinational circuit that accepts a 4-bit number and generates a 3-bit binary number output that approximates the square root of the number. For example, if the square root is 3.5 or larger, give a result of 4. If the square root is <3.5 and ≥ 2.5, give a result of 3.
3. In digital computer, letters of alphabet are coded in form of unique combination of five or more bits. One of the most common code is the six-bit ASCII code. The code is given in the following table in terms of octal equivalent. For example, $C=(03)_8$ = 000 011.

A	01	N	16
B	02	O	17
C	03	P	20
D	04	Q	21
E	05	R	22
F	06	S	23
G	07	T	24
H	10	U	25
I	11	V	26
J	12	W	27
K	13	X	30
L	14	Y	31
M	15	Z	32

Design a minimal sum-of-products circuit that will receive this code as an input and produce an output '1' whenever the letter is a vowel. The alphabet uses only 26 of possible codes and others are used for numerals and punctuation marks. However, you may assume that the data are pure alphabetic, so that only the code listed will occur.

4. The four lines into combinational logic circuit depicted in the figure below will carry one binary-coded decimal digit, i.e., decimal number 10-15 will never appear on the lines. The single output Z of the circuit is to be 1 if and only if the inputs represent a number that is either 0 or a power of 2. Construct the logic bloc diagram of a minimal two level realization of the circuit using only NAND gates.



5. Design a 4-bit adder using four full adders.
6. Design a 4-bit adder-subtracter using the instances of above 4-bit adders and other necessary gates.
7. Design a 12-bit adder-subtracter using the instances of above 4-bit adder-subtracters.
8. Design 1-digit BCD adder using the instances of above 4-bit adders and other necessary gates.
9. Design 4-digit BCD adder using the instances of above 1-digit BCD adders.
10. Design a combinational circuit to convert a 4-bit BCD number to the corresponding excess-3 code.
11. Design 1-digit excess-3 code adder using the instances of above 4-bit adders and other necessary gates.