**Electronics I Worksheet 4**

1. Simplify the following Boolean functions, using three-variable maps:

(a) F(x, y, z) = Σ(0, 2, 4, 5)

(b) F(x, y, z) = Σ(0, 2, 4, 5, 6)

1. Simplify the following Boolean expressions, using four-variable maps:

(a) A’B’C’D’ + AC’D’ + B’CD’ + A’BCD + BC’D

(b) x’z + w’ xy’ + w(x’y + xy’)

1. Find the minterms of the following Boolean expressions by first plotting each function in a map:

(a) xy + yz + xy’ z

(b) C’D + ABC’ + ABD’ + A’B’D

1. Convert the following Boolean function from a sum-of-products form to a simplified product-of-sums form. F(x, y, z) = Σ(0, 1, 2, 5, 8, 10, 13)
2. Simplify the following Boolean functions:

(a) F(A, B, C, D) = ∏(1, 3, 5, 7, 13, 15)

(b)F(A, B, C, D) = ∏(1, 3, 6, 9, 11, 12, 14)

1. Simplify the following Boolean function F , together with the don’t-care conditions d , and then express the simplified function in sum-of-minterms form:

(a) F(x, y, z) = Σ(0, 1, 4, 5, 6) and d(x, y, z) = Σ(2, 3, 7)

(b) F (A, B, C, D) = Σ (0, 6, 8, 13, 14) and d(A, B, C, D) = Σ (2, 4, 10)

1. Simplify the following functions, and implement them with two-level NAND gate circuits:

(a) F(A, B, C, D) = AC’D’ + A’C + ABC + AB’C + A’C’D’

(b) F(A, B, C, D) = A’B’C’D + CD + AC’D

1. Draw a logic diagram using only two-input NOR gates to implement the following function: F(A, B, C, D) = (A xor B)'(C xor D)
2. Derive the circuits for a three-bit parity generator and four-bit parity checker using an odd parity bit.
3. Consider the combinational circuit shown in Figure below



1. (a) Derive the Boolean expressions for T1 through T4. Evaluate the outputs F1 and F2 as a function of the four inputs. (b) List the truth table with 16 binary combinations of the four input variables. Then list the binary values for T1 through T4 and outputs F1 and F2 in the table. (c) Plot the output Boolean functions obtained in part (b) on maps and show that the simplified Boolean expressions are equivalent to the ones obtained in part (a).
2. Obtain the simplified Boolean expressions for output *F* and *G* in terms of the input variables in the circuit of Fig below and obtain the simplified Boolean expressions for output *F* and *G* in terms of the input variables in the circuit of Fig.



1. Design a combinational circuit with three inputs, *x* , *y* , and *z* , and three outputs, *A, B* , and *C*. When the binary input is 0, 1, 2, or 3, the binary output is one greater than the input. When the binary input is 4, 5, 6, or 7, the binary output is two less than the input.
2. A majority circuit is a combinational circuit whose output is equal to 1 if the input variables have more 1’s than 0’s. The output is 0 otherwise. Design a 3-input majority circuit by finding the circuit’s truth table, Boolean equation, and a logic diagram.
3. Design a combinational circuit that converts a four-bit Gray code to a bit four binary number. Implement the circuit with exclusive-OR gates.
4. Design a four-bit combinational circuit 2’s complementer. (The output generates the 2’s complement of the input binary number.) Show that the circuit can be constructed with exclusive-OR gates. Can you predict what the output functions are for a five-bit 2’s complementer?