COMP311 Extra Practice 2 Name: ______________________________

Question 1: Kmaps - Unsigned Comparison Assume A and B are two-bit unsigned numbers. We want to build a circuit that performs Y = 1 if A > B. Fill in this truth table.

A_1	A_0	B_1	B_0	**Y**

1.1. Write the boolean equation equivalent of this truth table.

Question 2: Kmaps - Simplification Fill out the kmap using the boolean expression that you have created.

AB \ CD	00	01	11	10
00				
01				
11				
10				

2.1. Write the equivalent simplified Boolean equation of this kmap.

Question 3: Practice Problem 1: Truth Table to K-Map to SOP Consider the 4-variable Boolean function $F(A, B, C, D)$ given below. X denotes a don't care condition.

A	B	C	D	F
0	0	0	0	0
0	0	0	1	1
0	0	1	0	X
0	0	1	1	1
0	1	0	0	0
0	1	0	1	X
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	X
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	X

3.1. Fill in the Karnaugh map below using the truth table.

AB \ CD	00	01	11	10
00				
01				
11				
10				

3.2. Using the don't cares, form the largest possible groups and derive the simplified Sum-of-Products expression.

Question 4: Practice Problem 2: Boolean Expression to K-Map to SOP Given the Boolean function:

$$F(A, B, C, D) = A\overline{B}C + \overline{A}BD + ABC + ACD$$

4.1. Fill in the Karnaugh map below.

AB \ CD	00	01	11	10
00				
01				
11				
10				

4.2. Simplify the function to its minimal Sum-of-Products form.

Question 5: Problem 1: K-Map Simplification Provide the equation below the K-map.

CD \ AB	00	01	11	10
00	1	0	1	1
01	0	0	1	1
11	1	1	0	0
10	1	0	0	1

Question 6: Problem 2: K-Map with Don't Cares Provide the equation below the K-map.

CD \ AB	00	01	11	10
00	1	0	1	1
01	0	0	1	1
11	1	1	0	0
10	1	0	0	1

Question 7: The Adaptive Control Unit You are designing the combinational control logic for a small embedded device used in a hazardous environment. The controller reads four sensor inputs, S_3, S_2, S_1, S_0 (with S_3 highest priority in normal operation). The device produces three outputs, A (alarm/valid) and L_1L_0 (2-bit encoded location). There is also a configuration bit, C (mode select).

Encoding convention for L_1L_0:

$$S_0 \mapsto 00, \quad S_1 \mapsto 01, \quad S_2 \mapsto 10, \quad S_3 \mapsto 11.$$

Alarm convention for A: $A = 1$ iff at least one sensor is active; otherwise $A = 0$.

Normal priority: $S_3 > S_2 > S_1 > S_0$
Diagnostic priority (reversed): $S_0 > S_1 > S_2 > S_3$

7.1. **Part I: Priority Encoding Core (Normal Mode)**

Assume the device is operating in **normal mode** (i.e., $C = 0$). The outputs must correspond to the *highest-priority active* sensor using the normal priority order.

7.1. Construct the complete truth table for the outputs A, L_1, and L_0 as functions of S_3, S_2, S_1, S_0 under normal priority.

7.2. Use K-maps to minimize the Boolean expressions for L_1 and L_0 in normal mode **without using any don't-cares**.

7.3. Implement the normal-mode encoder using **only** AND/OR/NOT gates.

7.4. Compute the total count of 2-input gates used in your implementation (count NOT gates separately) and the number of transistors used.

7.5. **Part II: Fault Masking and Don't-Care Optimization**

Engineering notes reveal the following input states **cannot physically occur** in the field:

$$0101, \quad 0110, \quad 0000$$

Treat these states as **don't-cares (X)**.

7.1. Redraw your K-maps for L_1 and L_0 using the don't-cares above. Then produce minimized Boolean expressions for L_1 and L_0.

7.2. Minimize A using the same don't-cares. Write the minimized Boolean expression.

7.3. Compare your *pre-optimization* and *post-optimization* circuits:

- Give the gate counts (2-input AND/OR + NOT) for each.
- Identify which signals/terms disappeared due to don't-cares.
- Compare the number of required transistors.

7.4. Explain conceptually **why** don't-cares can reduce hardware cost, and what assumption you are making about the real-world system when you use them.

7.5. **Part III: Configurable Architecture Using Multiplexers**

Now the controller must operate in **two modes** selected by C:

- **Normal mode** ($C = 0$): outputs behave according to Parts I–II (normal priority).
- **Diagnostic mode** ($C = 1$):
  - A must equal $S_3 \vee S_2 \vee S_1 \vee S_0$.
  - $L_1 L_0$ must encode the *lowest-priority active sensor* under the *reversed* priority $S_0 > S_1 > S_2 > S_3$ (encoding unchanged).

7.1. Construct the truth table for L_1 and L_0 in **diagnostic mode**.

7.2. Derive minimized Boolean expressions for L_1 and L_0 in diagnostic mode.

7.3. Design a combined circuit that outputs (A, L_1, L_0) for both modes by using **multiplexers as selectors** (i.e., selecting between normal-mode logic and diagnostic-mode logic). Provide a block diagram showing:

- what signals feed each MUX input,
- which MUX select line(s) are used,
- where any shared intermediate signals are reused.

7.4. **Optimization challenge:** Implement the entire system using **only** logic gates and multiplexers, and optionally **at most one** priority encoder block. If you use a priority encoder block, describe precisely how its inputs are wired in each mode (and how multiplexers accomplish this).

7.5. **Food for thought:** Suppose a future hardware revision makes one or more of the "impossible" states from Part II actually possible. Explain **one specific** incorrect behavior that could occur in your optimized design and **why** it happens.

Question 8: The Bank Vault Problem There are three employees at a bank: the Manager (M), the Cashier (C), and the Guard (G). The vault door opens only if:

- The Manager and at least one other person are present, OR
- Both the Guard and the Cashier are present.

8.1. Construct the truth table for the system.

8.2. Write the Sum of Products (SOP) expression.

8.3. Derive the simplified Boolean equation.

8.4. Draw the logic circuit using gates that minimize the transistor count.

8.5. Calculate the total number of transistors used in your final design.