1. (6 points)
   For the following Boolean function
   \[ f(a,b,c,d) = ac'd' + a'b'd + a'cd \]
   with don’t care conditions:  \[ a'c'd' + a'bc'd + acd + ab'cd' \]
   (1) Find all the prime implicants and essential prime implicants.
   (2) Simplify the function in the format of **sum of products** with minimum number of literals. (No need to draw the circuit.)

2. (4 points)
   (1) What is a half adder and what is a full adder?
   (2) Implement a full adder with half adders only. Draw the logic diagram.

3. (5 points)
   A 4-bit comparator is a system that takes two 4-bit numbers A and B as input and compares their values. Design a 4-bit comparator that has three output signals: G, L, E such that G=1 if A>B, L=1 if A<B, and E=1 if A=B. You can use two 4-bit shift registers and necessary logic gates.

4. (5 points)
   The following sequential circuit with three T flip-flops is designed to repeat the binary sequence 0→1→3→7→6→4→0 when initially all the flip-flops have 0 as their content. Verify the correctness of this design.
Digital Logic

(1) PIs: \( c'd', a'c', a'd, cd, ab'd' \)
    \( \text{EPI: } c'd' \)

(2) \( f = c'd' + a'd \)

2. A half adder adds up two bits \( a \) and \( b \) by giving the sum \( s \) and carry out \( c \).

A full adder adds up two bits \( a \) and \( b \) and a carry-in \( (\text{cin}) \) input and gives the sum \( s \) and carry out \( (\text{cout}) \).

\[ a \rightarrow HA \rightarrow c \rightarrow s \]
\[ b \rightarrow HA \rightarrow s \]

3. \( A_0, A_1, A_2, A_3 \)

\( G \) \( (A_3A_2A_1A_0 > B_3B_2B_1B_0) \)

\( L \) \( (A_3A_2A_1A_0 < B_3B_2B_1B_0) \)

\( E \) \( (A_3A_2A_1A_0 = B_3B_2B_1B_0) \)
if $A_3 > B_3$ \((A_3B_3')\) \(A > B\) \(G = 1\). stop shifting \((L = 0, E = 0)\)

if $A_3 < B_3$ \((A_3'B_3)\) \(A < B\) \(L = 1\). stop shifting \((G = 0, E = 0)\)

if $A_3 = B_3$, shift to compare $A_2$ and $B_2$ at the next clock

when $A = B$, after 4 clock pulses. $E = 0, L = 0 \quad E = (E+L)' = 1$

\[\begin{array}{c|cccc|cccc|c}
\text{A} & \text{B} & \text{C} & \text{TA} & \text{TB} & \text{TC} & \text{A} & \text{B} & \text{C} \\
\hline
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\
0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\
1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 6 \\
1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 4 \\
1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0
\end{array}\]

the above table shows that with initial state 000 the system will repeat the binary sequence 0, 1, 3, 7, 6, 4.