Solve the following K-Map.

Review Problem 18
7-seg display in Verilog

- Verilog RTL: just describe what you want

```verilog
module seg7 (bcd, leds);
    input logic [3:0] bcd;
    output logic [6:0] leds;

always_comb begin
    case (bcd)
        // 3210
        6'b0000: leds = 7'b0111111;
        6'b0001: leds = 7'b0000110;
        6'b0010: leds = 7'b1011011;
        6'b0011: leds = 7'b1001111;
        6'b0100: leds = 7'b1100110;
        6'b0101: leds = 7'b1101101;
        6'b0110: leds = 7'b1111101;
        6'b0111: leds = 7'b0000111;
        6'b1000: leds = 7'b1111111;
        6'b1001: leds = 7'b1101111;
        default: leds = 7'bX;
    endcase
end
endmodule
```
Verilog – Simulation & Mapping to FPGAs

Circuit Diagrams – FFT Implementation

K-Maps – Simplification Techniques

Boolean Algebra – Math Form for Optimization

Truth Tables – Case-by-case Circuit Description

Review: Circuit Implementation Techniques
1. Understand the Problem
2. Formulate the Problem in terms of a truth table or other suitable design representation
draw block diagram or other picture
write down inputs (data, control) and outputs
what is the circuit supposed to do?

3. Choose Implementation Target
   truth table, Boolean Algebra, Verilog, etc.

4. Follow Implementation Procedure
   K-maps, Boolean Algebra, Quartus Synthesizers

Combination Logic Design Process
Assume sensor reads "1" when tripped, "0" otherwise.
Inputs are three sensors, outputs are two arm control signals.

Understanding the Problem

Design combinational logic to activate the arms.

3 light barriers (light source + photocell) as sensors:
Rods too short stay on belt
Second arm pushes rods too long to other side
Mechanical arm pushes rods within spec (±5%) to one side
Rods of varying length (±10%) travel on conveyor belt

Statement of the Problem

Process Line Control Example
Where to place the light sensors A, B, and C to distinguish among the three cases?

Assume that A detects the leading edge of the rod on the conveyor
Process Line Control Example (cont.)

A to B distance place apart at specification - 5%
A to C distance placed apart at specification +5%
Process Line Control Example (cont.)

Rule: All values must be spaced by 2 sec + 5%
Similar to the main computation unit in a Microprocessor

\[ F = A \overline{B} \overline{C} + A \overline{B} \overline{C} + A \overline{B} \overline{C} \]

Create a unit that can compute the AND, OR, or XOR of two inputs

Logical Function Unit

A and B, based upon control lines C0 and C1
Logical Function Unit (cont.)

Implementation