\[ F = BC + AC \]

Solve the following K-Map.

Review Problem 18
module seven
endmodule

case (bcd) begin
always comp begin

input logic [6:0] bcd;
input logic [3:0] bc;
output logic [6:0] comp;

module seven (bcd, bc); //

endmodule

endcase
end

module seven
endmodule

always comp begin

always comp begin

//

endcase
end
Review: Circuit Implementation Techniques

- Truth Tables - Case-by-case circuit description
- Boolean Algebra - Math form for optimization
- K-Maps - Simplification technique
- Circuit Diagrams - TTL Implementations
- Verilog - Simulation & Mapping to FPGAs
1. Understand the Problem
2. Formulate the Problem in terms of a truth table or other suitable design representation
draw block diagrams or other pictures
write down inputs (data, control) and outputs
what is the circuit supposed to do?

3. Choose Implementation Target
   - k-maps, Boolean Algebra, Quarts synthesis
   - truth table, Boolean Algebra, Verilog, etc.

4. Follow Implementation Procedure
Draw a picture:

Call sensors A, B, C

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 + phototcell) 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
Assume that A detects the leading edge of the rod on the conveyor. Where to place the light sensors A, B, and C to distinguish among the three cases?

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

A to B distance place apart at specification - 5%

Process Line Control Example (cont.)

[Diagram showing distances and specification tolerances]
Process Line Control Example (cont.)
Logical Function Unit

Create a unit that can compute the AND, OR, or XOR of two inputs \( A \) and \( B \), based upon control lines \( C_0 \) and \( C_1 \).

Out = \( C_1 \overline{AB} + \overline{C_1} \overline{AB} \)

Similar to the main computation unit in a Microprocessor
Implementation:

Logical Function Unit (cont.)