Review Problem 17

- Solve the following K-Map.

\[ F = \overline{C} \overline{D} + \overline{B} \overline{D} \]
Case Study: Seven Segment Display

- Chip to drive digital display

<table>
<thead>
<tr>
<th>B3</th>
<th>B2</th>
<th>B1</th>
<th>B0</th>
<th>Val</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>9</td>
</tr>
</tbody>
</table>
Case Study (cont.)

- Implement L5:

<table>
<thead>
<tr>
<th>B3</th>
<th>B2</th>
<th>B1</th>
<th>B0</th>
<th>L5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

\[ L_5 = B_2 B_0 + \bar{B}_1 B_0 + B_2 \bar{B}_1 + B_3 \]
7-seg display in Verilog

- Verilog RTL: just describe what you want

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
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
Combinational Logic Design Process

1. Understand the Problem
   what is the circuit supposed to do?
   write down inputs (data, control) and outputs
   draw block diagram or other picture

2. Formulate the Problem in terms of a truth table or other suitable design representation
   truth table, Boolean Algebra, Verilog, etc.

3. Choose Implementation Target

4. Follow Implementation Procedure
   K-maps, Boolean algebra, Quartus synthesis
Process Line Control Example

Statement of the Problem

- Rods of varying length (+/-10%) travel on conveyor belt
- Mechanical arm pushes rods within spec (+/-5%) to one side
- Second arm pushes rods too long to other side
- Rods too short stay on belt

- 3 light barriers (light source + photocell) as sensors
- Design combinational logic to activate the arms

Understanding the Problem

- Inputs are three sensors, outputs are two arm control signals
- Assume sensor reads "1" when tripped, "0" otherwise
- Call sensors A, B, C
- Draw a picture!
Process Line Control Example (cont.)

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%