BCD to 7-segment display controller

Understanding the problem
- input is a 4 bit bcd digit (A, B, C, D)
- output is the control signals for the display (7 outputs C0 – C6)

Block diagram

Formalize the problem

Truth table
- show don’t cares

Choose implementation target
- if ROM, we are done
- don’t cares imply PAL/PLA may be attractive

Follow implementation procedure
- minimization using K-maps
Implementation as minimized sum-of-products

- 15 unique product terms when minimized individually

\[
\begin{align*}
C_0 &= A + B D + C + B' D' \\
C_1 &= C' D' + C D + B' \\
C_2 &= B + C' + D \\
C_3 &= B' D' + C D' + B C' D + B' C \\
C_4 &= B' D' + C D' \\
C_5 &= A + C' D' + B D' + B C' \\
C_6 &= A + C D' + B C' + B' C
\end{align*}
\]

Implementation as minimized S-o-P (cont'd)

- Can do better
  - 9 unique product terms (instead of 15)
  - Share terms among outputs
  - Each output not necessarily in minimized form

\[
\begin{align*}
C_0 &= A + B D + C + B' D' \\
C_1 &= C' D' + C D + B' \\
C_2 &= B + C' + D \\
C_3 &= B' D' + C D' + B C' D + B' C \\
C_4 &= B' D' + C D' \\
C_5 &= A + C' D' + B D' + B C' \\
C_6 &= A + C D' + B C' + B' C
\end{align*}
\]
### PLA implementation

- **A**, **B**, **C**, **D**
- Inputs:

### PAL implementation

- Limit of 4 product terms per output
- Decomposition of functions with larger number of terms
- Do not share terms in PAL anyway (although there are some with some shared terms)
  
### PAL implementation

- Find common sub-expressions among functions

---

5/2/2001 CSE 370 - Spring 2001 - Combinational Implementation - 5

5/2/2001 CSE 370 - Spring 2001 - Combinational Implementation - 6
**Production line control**

- 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 that are 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

**Sketch of problem**

- Position of sensors
  - A to B distance = specification – 5%
  - A to C distance = specification + 5%
Formalize the problem

Truth table

(show don't cares)

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>Function</th>
<th>logic implementation now straightforward</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>do nothing</td>
<td>just use three 3-input AND gates</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>do nothing</td>
<td>&quot;too short&quot; = AB'C'</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>do nothing</td>
<td>(only first sensor tripped)</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>do nothing</td>
<td>&quot;in spec&quot; = A B C'</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>too short</td>
<td>(first two sensors tripped)</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>don't care</td>
<td>&quot;too long&quot; = A B C</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>in spec</td>
<td>(all three sensors tripped)</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>too long</td>
<td></td>
</tr>
</tbody>
</table>