Miniproject 1
A new thermostat system for Gulley Hall

Due to a contractor going bankrupt and the discomfort of certain administrators, proposals were solicited for finishing up the temperature control system for Gulley Hall. To the surprise of no one (and the chagrin of many), the low bidder was Robertics, Inc., who has recently “redefined the state of the art” in communications systems. The temperature control system is nearly complete, but the odd use of temperature data made most bidders wary of taking on the project at this late date (especially given the extremely short deadline).

The system behavior is fairly simple. The 4 system modes are heating, cooling, neither heating nor cooling, and both heating and cooling. The interface for setting the modes is in place, the mode information is available as 2 signals, heating and cooling. The desired temperature is set as two thresholds, Tl and Tu, to avoid rapid cycling (Tl is the desired temperature - delta, Tu is the desired temperature + delta, where delta is a couple of degrees). The existing circuitry calculates these values, and compares them with the current temperature. The results of these comparisons are made available as logic signals Tc<Tl (room temperature less than Tl), and Tc<Tu (room temperature less than Tu).

An “interesting” feature of the existing circuitry is that the temperature comparison signals are only intermittently available. There is a signal /T-ok that is active when Tc<Tl and Tc<Tu have legitimate (correct) values. According to the specification, whenever /T-ok becomes active, it will stay active for at least three clock cycles (a clock signal is also available), and the signal should be available at least a few times a minute. The clock runs at some undetermined speed in the 10 kHz range.

The proposed system will be implemented as a Mealy machine (outputs depend on states and inputs); the outputs are cooling-on and heating-on. The behavior is explained in the following table:

<table>
<thead>
<tr>
<th>Temperature</th>
<th>Mode: Heating</th>
<th>Mode: Cooling</th>
<th>Mode: Neither</th>
<th>Mode: Both</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cold</td>
<td>Heat on/Cool off</td>
<td>Heat off/Cool off</td>
<td>Both off</td>
<td>Heat on/Cool off</td>
</tr>
<tr>
<td>Medium</td>
<td>Unchanged</td>
<td>Unchanged</td>
<td>Both off</td>
<td>Both off</td>
</tr>
<tr>
<td>Hot</td>
<td>Heat off/Cool off</td>
<td>Heat off/Cool on</td>
<td>Both off</td>
<td>Heat off/Cool on</td>
</tr>
</tbody>
</table>

where Cold means the last temperature reading had the room temperature less than Tl, Medium means it was between Tl and Tu, and Hot means it was greater than or equal to Tu. The “interesting” behavior takes place when the temperature is in the medium range: in Heating and Cooling modes, the outputs should not change, (in “Both” mode, the outputs should be set inactive). You will deal with this by having two different states for Medium: Medium-from-Cold, and Medium-from-Hot, allowing you to know which non-medium state you were in previously. (You will need to figure out why this helps!).

Your job is to design the Mealy machine that will provide this control. Step zero, coming up with a state-transition diagram, is made easier by Figure 1, where states correspond to the last temperature reading available, (except for the two in the middle: both of these have Medium as the last temperature available). Sadly, the transitions are not labelled as to what inputs cause the transitions, and I have left off the transitions from a state to itself, but this should provide a good starting point. *Your design should use these states!* The procedures given on p.563 of Wakerley provide good guidance on how you should proceed.
Deliverables

The inputs to your system are the following variables (defined above): heating, cooling, /T-ok, Tc<Tl, Tc<Tu, and clock. The outputs are heating-on and cooling-on. You may wish to provide a synchronous or asynchronous reset input to ensure that the system can be put into a reasonable state.

Go through the steps on p. 563, including labeling the edges on the Figure 1 diagram. Explain how state variable combinations map to states. Provide all of the various tables and equations. Draw and test the circuit in LogicWorks.

Due date: April 6 (!), in discussion section.

This is only one week, so you need to get going now, even if the lectures have not covered all you need. For any help, ask Robert or Gamal. It is an extremely good idea to read the material in Sections 7.3 and 7.4 immediately, so you can put together a preliminary design (state diagram, transition/output table, and state assignment) and be sure you know what to do by Thursday, so there is time to get help if you need it.