Summer 2011 CSE 140L
Lab 1 Assignment
UCSD Alarm Clock

Alex Orailoglu
06/27/11
Behavioral Description

• **Part 1: Basic alarm clock**
  
  – User can adjust the clock time and the alarm time.
  – A single set of LCDs that display
    • The alarm time when the user is setting the alarm;
    • The current time in all other cases.
  – User can turn the alarm on or off.
  – If the alarm is on, the buzzer goes high when
    • clock hour = alarm hour, and
    • clock minutes = alarm minutes.
Behavioral Description

• **Part 1: Basic alarm clock**
• **Part 2a: Extending buzzer hold time**
  – In part 1, the buzzer stays on only for 1 minute.
    • Too short to wake some of us up
  – In this part, you should extend the buzzer hold time so that
    • The buzzer will keep buzzing unless the user turns it off
    • The buzzer goes high when
      – clock hour = alarm hour,
      – clock minute = alarm minute, and
      – Alarm is on
    • The buzzer goes off when
      – Alarm is turned off
    • Once turned off, turning the alarm back on does NOT reactivate the buzzer
Behavioral Description

• **Part 1: Basic alarm clock**

• **Part 2a: Extending buzzer hold time**

• **Part 2b: One hour buzz scheme**
  
  – In part 2a, the buzzer stays on *forever* if nobody turns the alarm off: very annoying in some situations.
  
  – In this part, you should design a one hour buzz scheme so that
    
    • The buzzer can keep on for one hour if the alarm is on
    • Turning the alarm off directly deactivates the buzzer
    • Once turned off, turning the alarm back on, even within the one hour period, does NOT reactivate the buzzer
Behavioral Description

• **Part 1: Basic alarm clock**

• **Part 2: Buzzer Hold Time**

• **Part 3a: Differentiating between MW vs. TuTh alarm times**
  - The alarm clock keeps track of the day information, and displays it on 7-seg Disp I/O
  - User can adjust the day
  - Two alarm times
    - One for **MW** and one for **TuTh**
    - Alarm clock automatically selects the effective alarm time for buzzing based on the day of the week information
    - Each alarm can be turned on/off and adjusted separately

• **Part 3b: Disabling Extended Weekend Alarm**
  - Alarms for the **extended weekend** automatically disabled
Components to implement Part 1

- **ClockDev:**
  - Seconds are incremented at each CLK cycle as in a regular clock. When its Set input is set, clock time can be adjusted by incrementing hours and minutes using SM (Set Minute) and SH (Set Hour) inputs.

- **AlarmDev:**
  - Very similar to ClockDev except that this block has neither a CLK input nor output for seconds.

- **LCD Interface:**
  - Takes a 6-bit input number (0 to 59) and displays it on 7-segment displays as two decimal digits.

- **6Comp:**
  - 6-bit comparator that gives a ‘1’ output if the two inputs are the same.

- **MUX-2x6:**
  - 6-bit 2-to-1 multiplexer that selects one of the two 6-bit inputs and directs it to its output.

- **Decoder-4 (LogicWorks library component):**
  - If En == 1, all output = 1;
  - If En == 0, For i = 0 to 3, Qi = 0 if S1S0 == i;
Components for Parts 2a & 2b

- **D Flip Flop (LogicWorks library component):**
  - If R (reset) = 0, Q = 0.
  - If S (set) = 0, Q = 1.
  - If R = 1 and S = 1, whenever C (clock) changes from 0 to 1,
    - Q = D ;
  - Q retains its current value in other cases.
  - This component can be used to hold the buzzer state.

- **Hints for implementing correct buzzer hold time:**
  - Map the alarm behavior to the functionality of flip-flops using simple glue logic (one gate)
Components for Part 3

- **DayCounter**
  - A modulo-7 counter, counting from 0 to 6, then back to 0
  - Encoding of days:

<table>
<thead>
<tr>
<th>Monday</th>
<th>Tuesday</th>
<th>Wednesday</th>
<th>Thursday</th>
<th>Friday</th>
<th>Saturday</th>
<th>Sunday</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
</tbody>
</table>

- If CLR = 1, counter is reset to 0 (Monday)
- If CLR = 0, counter increments by 1 whenever INC changes from 0 to 1