

University of California at Berkeley
College of Engineering
Department of Electrical Engineering and Computer Science

EECS 150
Fall 2005

R. H. Katz

Problem Set # 6
(Assigned 12 October, Due 21 October)

1. Consider the design of a simple digital alarm clock. The clock displays the current time in 24-hour HH:MM notation (HH goes from 00 to 23 and MM from 00 to 59). There are several control inputs for setting the current time and the alarm time. The TIME SET HH button allows you to rapidly advance through the hours 0 through 23. The time advances one hour for each time the button is pressed. The TIME SET MM button does the same thing for the minutes 00 through 59. In addition, the clock has a toggle input that indicates CLOCK in one position and ALARM in the other. This refers to which time is being set by the TIME SET buttons: either the current time (CLOCK) or the alarm time (ALARM). If the ALARM button is set, the clock displays the last set alarm time. Finally, there is another input controlled by a push button: ALARM ON when pushed in and ALARM OFF when popped out. Your job is to define the state diagram for the clock controller. It should support normal clock display (when CLOCK is set), alarm display (when ALARM is set), set current time (CLOCK + TIME SET buttons), set alarm time (ALARM + TIME SET buttons), and of course, the alarm should sound when the current time equals alarm time.
 - (a) Identify your inputs, outputs, and name and describe your states. State any assumptions you are making.
 - (b) Draw a symbolic state diagram for your design, labeling all state transitions.
 - (c) Write sketch Verilog for describing this state machine, assuming a MOORE Machine implementation.

2. Design an in-car “trip computer” to the following specification. Under driver control, the trip computer displays miles traveled since last reset, average speed (miles traveled/elapsed time since last reset), and miles per gallon (miles traveled/gallons consumed since last reset). The inputs to the state machine are: (1) signal to increment miles traveled (once per mile traveled), (2) increment time (once per second), and (3) the integer number of gallons of gas consumed since the last reset. The user interface has two buttons: a *reset* and a *display* push button. By default, miles traveled since last reset is displayed. If *display* is pressed, the average speed in *miles per hour* is shown. If pressed again, the *miles per gallon* is shown. Pressed one more time, the display returns to miles traveled since last reset, and the display sequence can be repeated. You may assume that *display* is a synchronized, debounced signal that is asserted for exactly one clock period when pressed.
 - (a) Draw a simple block diagram that shows the controller’s inputs and outputs. Document your assumptions.
 - (b) Draw a simple datapath block diagram for the trip computer, including any multiplexers, counters, registers, arithmetic functional units (e.g., adders, subtractors, multipliers, dividers), and display decoders and drivers you may need for its implementation. Don’t worry about how wide these functional units are – simply assume that they are designed to be wide enough for this application. Show how they are interconnected and indicate the control signals that you need to control this datapath.
 - (c) Draw a complete MEALY MACHINE State Diagram for your controller.
 - (d) Write sketch Verilog for describing the state machine.